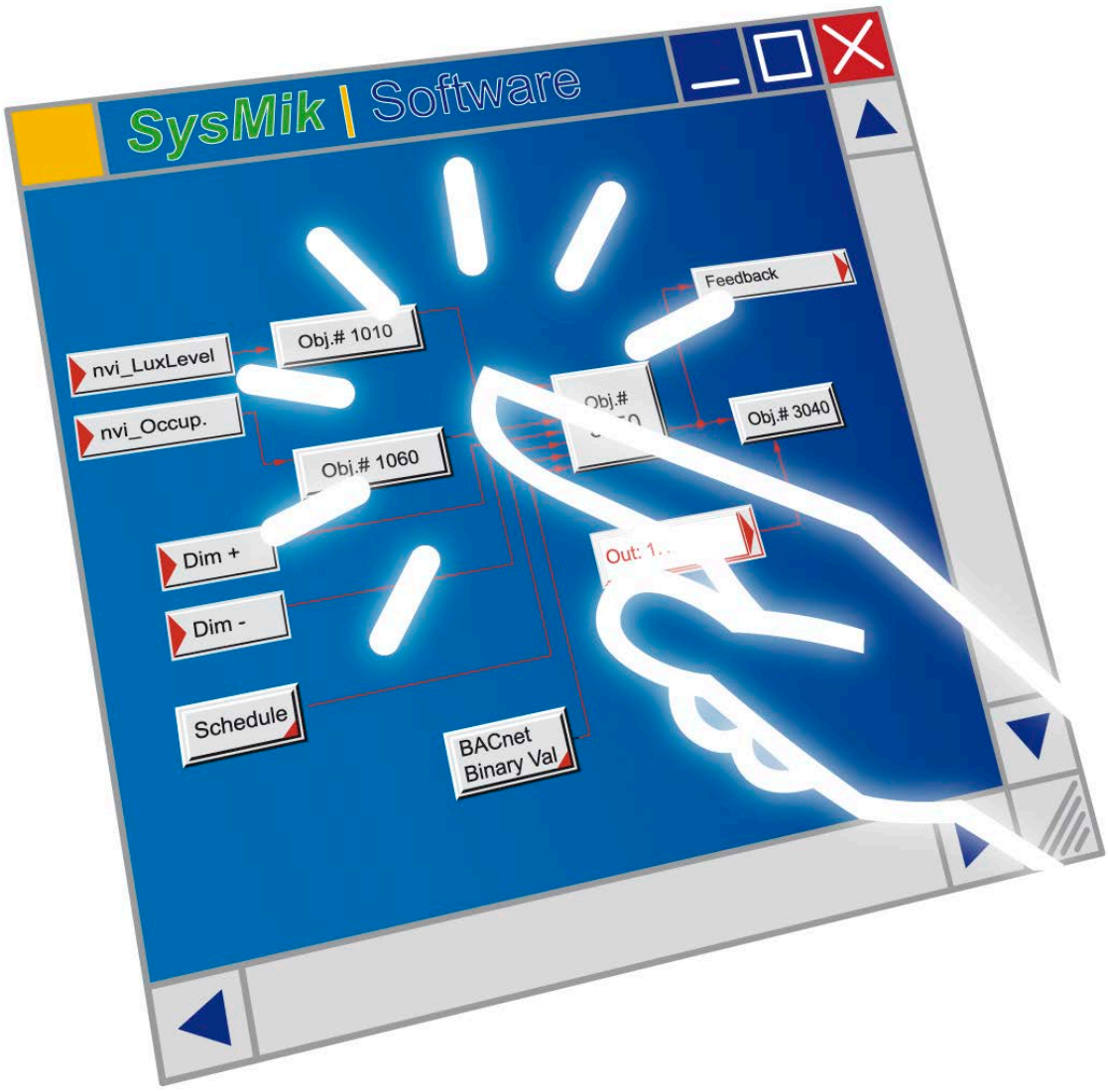
Erste Schritte in IPOCS
Version 4.1.12

Bedienungsanleitung



Revision 3

SysMik
GmbH Dresden

Diese Anleitung unterstützt Sie bei Einbau und Nutzung des Gerätes. Die Informationen wurden mit größter Sorgfalt zusammengestellt. Wir haben den Inhalt des Dokumentes anhand der beschriebenen Hardware und Software sorgfältig geprüft, können eventuelle Abweichungen jedoch nicht ausschließen. Deshalb übernehmen wir keine Haftung für mögliche Fehler, die in dieser Beschreibung und in ggf. mitgelieferter Beispielsoftware enthalten sein könnten. Änderungen der Geräte sowie der zugehörigen Dokumente bleiben vorbehalten.

Für Kritik und Anregungen sind wir Ihnen dankbar. Nähere Informationen, wie weiterführende Beschreibungen, Ausschreibungstexte zu Geräten und über verfügbare Software, finden Sie im Internet unter www.sysmik.de. Auf Wunsch senden wir Ihnen diese gern zu.

Die Garantie für das Gerät erlischt bei unsachgemäßer Handhabung, bei Gerätedemontage sowie bei Verwendung von nicht durch SysMik für dieses Gerät freigegebener Software. Inbetriebsetzung und der Betrieb des Gerätes darf nur unter Beachtung der gültigen Sicherheitsbestimmungen und durch qualifiziertes Personal vorgenommen werden.

SysMik® und das SysMik-Logo sind eingetragen Warenzeichen der SysMik GmbH Dresden. IPOCS ist ein Warenzeichen der SysMik GmbH Dresden. "Networking Together!"© unterliegt dem Copyright der SysMik GmbH Dresden. Echelon®, LON®, LONWORKS®, LonBuilder®, NodeBuilder®, LonManager®, LonTalk®, LonUsers®, LonPoint®, Neuron®, 3120®, 3150®, Digital Home®, LONMARK®, das Echelon Logo, das LONMARK Logo und das LonUsers Logo sind Warenzeichen der Echelon Corporation, registriert in den USA und anderen Ländern. LonLink®, LonResponse®, LonSupport®, LONews®, LonMaker® und i.LON® sind Warenzeichen der Echelon Corporation. BACnet® ist eingetragenes Warenzeichen der American Society of Heating, Refrigerating and Air Conditioning Engineers, Inc. (ASHRAE).

Alle anderen in dieser Anleitung gebrauchten Warenzeichen sind eingetragener Besitz der jeweiligen Eigentümer. Diese und weitere Warenzeichen sind im Text verwendet, werden jedoch im Interesse der Lesbarkeit im Weiteren nicht eigens gekennzeichnet.

Die Vervielfältigung, Weitergabe dieses Dokumentes, sowie die Verwertung und Mitteilung des Inhaltes ist nur mit Einverständnis der SysMik GmbH Dresden gestattet.

Copyright © 2012 by SysMik GmbH Dresden

SysMik GmbH Dresden	**Tel**	**+ 49 (0) 351 - 4 33 58 - 0**
Bertolt-Brecht-Allee 24	**Fax**	**+ 49 (0) 351 - 4 33 58 - 29**
01309 Dresden	**E-Mail (Verkauf)**	**sales@sysmik.de**
	E-Mail (Support)	**service@sysmik.de**
Germany	**Homepage**	**http://www.sysmik.de**

Inhalt

1 Einführung – Willkommen bei IPOCS-4.1 5

1.1 Systemvoraussetzungen 6

1.2 Die Installation von IPOCS-4.1 6

1.3 Was ist neu? 7

2 Programmieren der Applikation 8

2.1 Anlegen eines leeren Projektes 8

2.2 Hardware und Software 10

2.3 Schritte der Programmierung 12

3 Test und Inbetriebnahme 16

3.1 Test mit Hilfe der PC-Simulation 17

3.2 Download in ein reales Zielsystem 19

3.3 Schritte der Inbetriebnahme des realen Zielsystems 21

4 Netzwerkfunktionen 23

4.1 Erweiterung des Projektes um Netzwerkvariablen 23

4.2 LonMark-Unterstützung 27

4.3 Systemintegration 32

5 IPOCS für Inline Control Server 33

5.1 IO-Zugriff 34

5.2 Netzwerkvariablen 36

5.3 Verarbeitungsfunktionen 37

5.4 BACnet 39

5.5 Webserver 40

5.6 E-Mail 40

5.7	Systemfunktionen	40
5.8	Fidelio	40
5.9	DALI	40
5.10	EnOcean	41
5.11	Modbus	41
5.12	Zielsystemkerne	41
5.13	Systemdiagnose	42
5.14	IP-Kommunikationstreiber	42
5.15	Programmierbeispiele	43
6	Ausblick: Was ist noch möglich?	44
7	Literatur	49

1 Einführung – Willkommen bei IPOCS-4.1

IPOCS-4.1 – Interactive Programming of Open Control Systems ist ein grafisches Programmierwerkzeug für LonWorks-Geräte. Es wendet sich vorrangig an Anwender, die nicht über spezielles LonWorks-Know-How sowie tiefgründige Programmiererfahrungen verfügen. Es stellt alle Werkzeuge zur Programmierung sowie zur Simulation, Test und Download der Applikation in IPOCS-programmierbare LonWorks-Geräte zur Verfügung.

IPOCS-4.1 hat eine grafische Benutzeroberfläche. Die Software nutzt die grafischen Fähigkeiten des Windows-Betriebssystems und orientiert sich im Nutzerinterface an bekannten Windows-Techniken, so dass der Umgang mit dem Programm intuitiv und leicht erlernbar ist.

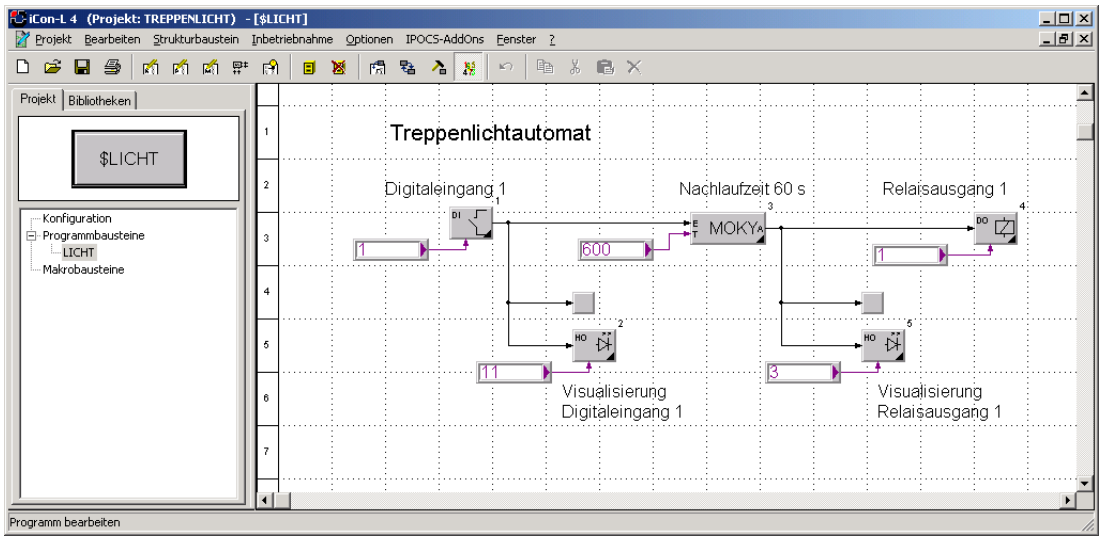


Bild 1.1: Beispiel für ein in IPOCS-4 geöffnetes Projekt

IPOCS-4.1 hat eine objektorientierte Struktur. Durch Verwendung einer objektorientierten Datenbasis, starke Modularisierung der einzelnen Baugruppen und Trennung von Applikationserstellung und Softwaretest sowie Download wird der Nutzer in den grundlegenden Phasen der Softwareentwicklung transparent unterstützt. Dabei werden keine starren Grenzen auferlegt und ein Wechsel zwischen den verschiedenen Betriebsmodi und -phasen ist jederzeit möglich.

IPOCS-4.1 ist flexibel. Mit IPOCS-4.1 können unterschiedliche Zielsysteme mit unterschiedlichen Funktionsbibliotheken programmiert werden. Somit ist eine Anpassung an spezifische Anwendungsfälle möglich.

IPOCS-4.1 ist vielseitig einsetzbar. Es existiert bereits eine breite Palette von IPOCS-programmierbaren Geräten (Zielsystemen) mit unterschiedlichsten Peripherieanschaltungen.

Funktionsbausteine sind das Herz. Die durch das jeweilige Zielsystem ausführbaren Funktionen sind in IPOCS-4.1 als Funktionsbausteine verfügbar. Durch die Verbindung beliebiger Funktionsbausteine entsteht eine IPOCS-Applikation. Die verschiedenen Funktionsbausteine sind nach Kategorien geordnet in Bausteinbibliotheken zusammengefasst.

Diese Anleitung soll an einem einfachen Beispiel zeigen, wie Sie mit IPOCS-4.1 Applikationen für LonWorks-Geräte entwickeln können.

1.1 Systemvoraussetzungen

PC mit Windows 2000, Windows XP oder Windows 7 *)

LON Netzwerk Schnittstelle z.B. von Echelon oder Loytec

*) Die Online-Hilfen liegen teilweise in einem älteren Format (hlp) vor. Um diese Hilfen unter Windows 7 anzeigen zu können, muss das Windows-Hilfe-Programm WinHlp32.exe vom Windows Download Center installiert werden.

Beschränkung / Autorisierung

Das Programmiersystem IPOCS-4.1 wird mit einem Dongle vor unbefugter Nutzung geschützt. Ohne Dongle (=IPOCS-Probeversion) ist ein IPOCS-Projekt nur zwei Stunden bearbeitbar. Weiterhin werden die ohne Dongle geladenen Applikationen nur zwei Stunden auf den Zielsystemen arbeiten.

1.2 Die Installation von IPOCS-4.1

Bevor Sie mit IPOCS-4.1 arbeiten können, muss das Programm auf Ihrem PC installiert werden. Dazu legen Sie die mitgelieferte CD-Rom in Ihren PC ein und starten das Installationsprogramm **setup.exe**. Bei der Installation folgen Sie bitte den Anweisungen des Installationsassistenten. Das Programm erstellt Verzeichnisse, kopiert die nötigen Dateien und legt die IPOCS-Programmgruppe in Ihrem Startmenü an.

Setup - IPOCS
IPOCS™
Herzlich Willkommen zum IPOCS Setup!
Dieses Installationsprogramm wird jetzt IPOCS 4.1.4 und zugehörige Komponenten auf Ihren Computer installieren.
"Weiter" zum Fortfahren, "Abbrechen" zum Verlassen.
www.sysmik.de
Weiter >
Abbrechen

Bild 1.2.1: Installationsassistent

1.3 Was ist neu?

IPOCS-4.1 ist vollständig abwärtskompatibel. Alle Projekte, die Sie mit früheren IPOCS-Versionen erstellt haben, können Sie problemlos mit IPOCS-4.1 weiter bearbeiten. IPOCS-4.1 bietet folgende Neuigkeiten:

Neuer LON Configurator

Der neue LON Configurator erhöht den Nutzerkomfort bei der Definition der LON-Netzwerkvariablen. Großer Wert wurde auf die Unterstützung bei der Erstellung LonMark-konformer Interfaces gelegt. So werden z.B. alle SNVTs, SCPTs und Standard Functional Profiles zur Nutzung angeboten. Über die Favoriten lassen sich häufig benutzte Elemente besonders schnell finden. Der neue LON Configurator kann über das Menü ***IPOCS- AddOns > Lon Configurator*** gestartet werden. Im Dokument „Erste Schritte mit dem LON Configurator“ wird die Bedienung ausführlich beschrieben.

Anwendungsbibliotheken für Raumautomation nach VDI 3813 Blatt 2

Es stehen zahlreiche Anwendungsfunktionen in zur Verfügung, mit denen typische Aufgaben der Gebäudeautomation gelöst werden können. Alle Funktionen sind als Makro (IPOCS Quelltext) implementiert und können bei Bedarf an spezielle Bedürfnisse angepasst werden. Die Bibliotheken incl. Online-Hilfe sind im Verzeichnis \Lib\Makro abgelegt.

Über das Menü ***Strukturbaustein > Exportieren…*** und ***Strukturbaustein Importieren...*** können die Makros in eigene IPOCS-Projekte übernommen und dort auch angepasst werden.

Über das Menü ***Projekt > Bibliothek laden…*** wird nur eine Referenz auf die Bibliothek erzeugt. Die Makros können unverändert verwendet werden; eine Beschreibung der Makros ist in der Online-Hilfe enthalten.

IPOCS WebDesigner

Die Visualisierung hat nun direkten Zugriff auf Datenpunkte der BACnet-, Modbus- und DALI-Schnittstellen.

Modbus RTU

ICS-Zielsysteme mit Ethernetschnittstelle unterstützen nun auch das Feldbusprotokoll Modbus RTU. Der ICS arbeitet dabei als Modbus RTU-Master, der über einen separat erhältlichen RS485-Adapter Datenpunkte fremder Modbus RTU-Slaves lesen und schreiben kann.

Port Mirroring

Der interne Ethernet-Switch der ICS-Zielsysteme kann bei Bedarf in die Betriebsart Port Mirroring versetzt werden. In diesem Modus arbeitet er wie ein Hub; d.h. eingehende Daten werden an alle Ports weitergeleitet. Dadurch ist es ohne zusätzliche Hilfsmittel möglich, die Ethernet-Kommunikation des ICS mit einem PC aufzuzeichnen und zu analysieren.

2 Programmieren der Applikation

2.1 Anlegen eines leeren Projektes

Starten Sie IPOCS-4.1 über das Startmenü: ***Start > Programme > SysMik > IPOCS4.1.8 > IPOCS4.1.8***.

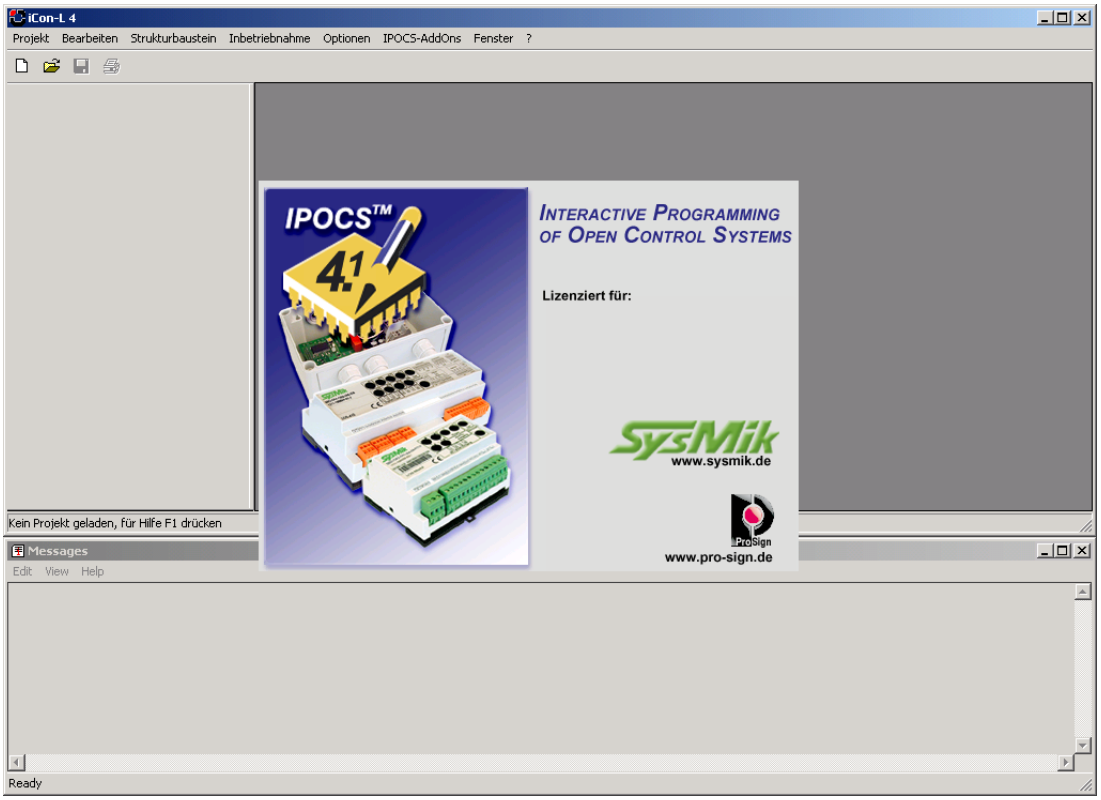


Bild 2.1.1: Startoberfläche von IPOCS-4.1

Von der Startoberfläche aus können Sie nun mit der Erstellung eines neuen Projektes beginnen. Dazu wählen Sie im Menü ***Projekt > Neu***.

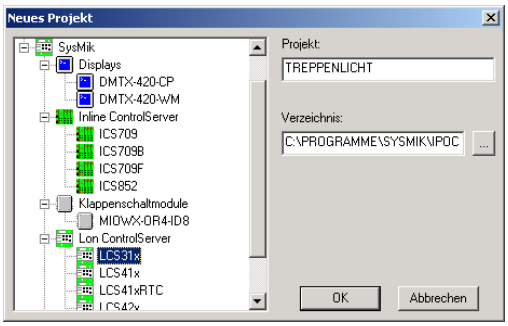


Bild 2.1.2: Dialog „Neues Projekt“

Im Dialog ***Neues Projekt*** können Sie den Namen des Projektes festlegen und das Projektverzeichnis (hier werden alle Daten des Projektes abgelegt) bestimmen. Weiterhin geben Sie an, für welches Zielsystem eine Applikation erstellt werden soll.

Ein IPOCS-Zielsystem besteht aus einem LonWorks-Gerät (Hardware) mit einem IPOCS-Zielsystemkern und einer ganzen Reihe von Funktionen, die durch die

Funktionsbausteine in IPOCS-4.1 zugreifbar sind. Je nach Ausrichtung des Zielsystems sind unterschiedliche Funktionsbausteine verfügbar. Durch die Auswahl des Zielsystems in diesem Dialog werden die zu verwendenden Bibliotheken mit den entsprechenden Funktionsbausteinen festgelegt.

Nach Bestätigung der Eingaben mit ***OK*** schaltet IPOCS-4.1 sofort in den Editiermodus.

Auf der linken Seite sehen Sie im Projektbaum die Projektstruktur: Hier werden neben der Projektkonfiguration alle Programmbausteine und Makrobausteine des Projektes aufgelistet.

Es existiert bereits ein leerer Programmbaustein **TASK1**, der im Arbeitsbereich als Arbeitsblatt angezeigt wird. Die Konfiguration des neuen Projektes (d.h. die Zusammenstellung der Applikation aus den verfügbaren Programmbausteinen) besteht aus genau einer Instanz dieses Programmbausteins. Somit können wir sofort damit beginnen, den Programmbaustein zu füllen.

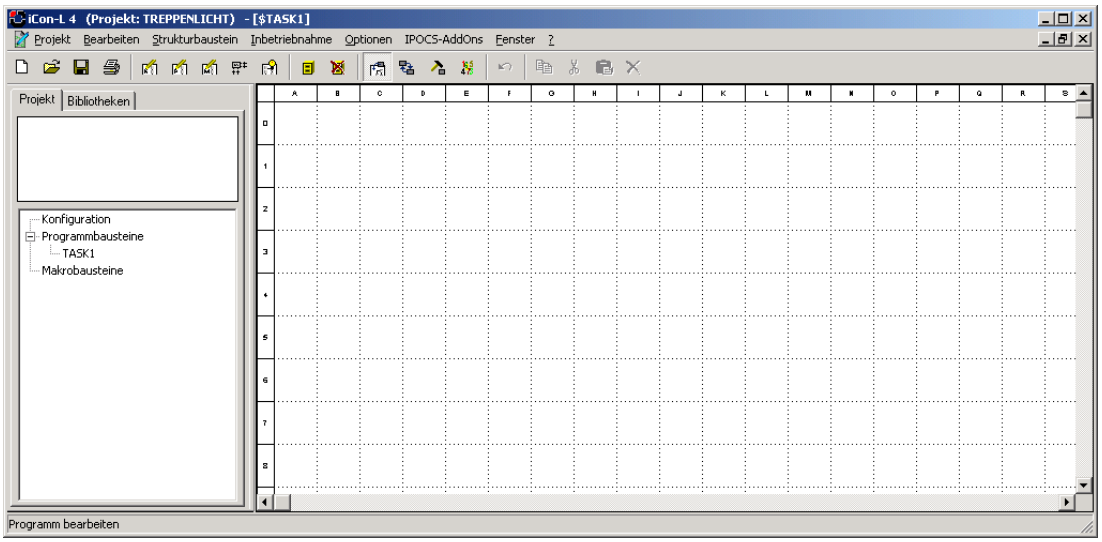


Bild 2.1.3: Leeres Projekt

Um die Dokumentation anschaulicher zu gestalten, werden die Arbeitsschritte an einem Beispiel erläutert. Die **Aufgabe 1** fasst die bis hierher ausgeführten Schritte zusammen.

Aufgabe 1: Erstellen Sie ein neues Projekt, geben Sie diesem Projekt den Namen „Treppenlicht“ und wählen Sie das Zielsystem „LCS310“ aus.

Wenn Sie diese Aufgabe ausgeführt haben, ist Ihnen der erste Schritt zur Erstellung eines eigenen Projektes gelungen.

2.2 Hardware und Software

Für den Einsatz der mit IPOCS-4.1 erstellten Applikation sind spezielle Geräte (Zielsysteme) notwendig. Die nun folgenden Beispiele werden an Hand des Gerätes LON-Control-Server LCS310 (SysMik GmbH Dresden) erklärt.

Alle in diesem Dokument erläuterten Funktionen sind auch für andere IPOCS-Zielsysteme nutzbar.

Damit ein Zielsystem eine Aufgabe bearbeiten kann, muss es über die entsprechende Peripherie (Hardwareein- und -ausgänge) Netzwerkvariablen und Algorithmen verfügen. Alle diese Funktionen werden als Funktionsbausteine in IPOCS-4.1 abgebildet.

Hardwareein- und -ausgänge

Das Gerät LCS310 besitzt:

- analoge Eingänge (z.B. Messen von Temperaturen oder Spannungen)
- digitale Eingänge (z.B. Anschluss von Schaltern; auch als S0-Zähler nutzbar)
- analoge Ausgänge (z.B. als Stellglied)
- digitale Ausgänge (Relais)

Diese I/O-Kanäle werden durch Funktionsbausteine der Bibliothek **Hardware I/O** verfügbar gemacht. Für jede Art von Eingängen / Ausgängen existiert ein Baustein, der über einen Parametereingang die Kanalnummer erhält. Die Zählung der Kanäle beginnt immer bei 1 und folgt der Kanalzählung der Gerätebeschreibung.

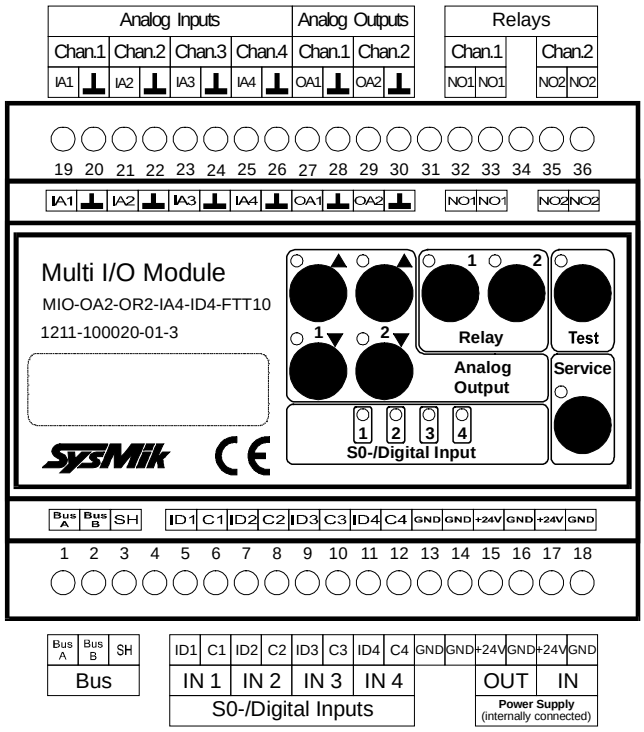

Analog Inputs
Analog Outputs
Relays
Chan.1 Chan.2 Chan.3 Chan.4 Chan.1 Chan.2 Chan.1 Chan.2
IA1 IA2 IA3 IA4 OA1 OA2 NO1 NO1 NO2 NO2
19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36
Multi I/O Module
MIO-OA2-OR2-IA4-ID4-FTT10
1211-100020-01-3
Relay
Test
Analog Output
Service
S0-/Digital Input
Bus A Bus B SH ID1 C1 ID2 C2 ID3 C3 ID4 C4 GND GND +24V GND +24V GND
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
Bus
IN 1 IN 2 IN 3 IN 4
S0-/Digital Inputs
OUT IN
Power Supply
(internally connected)

Bild 2.2.1: Strukturbild des LCS310

Hinweis:
Andere Zielsysteme können abweichende Hardwareein- und -ausgänge besitzen. Eine Beschreibung über die Gerätespezifikationen finden Sie in der beiliegenden Geräte-Dokumentation.

Darüber hinaus verfügt der LCS310 über eine eigene Testbedienebene aus LEDs und Tastern. Die Zuordnung der Kanalnummern können Sie folgender Abbildung entnehmen:

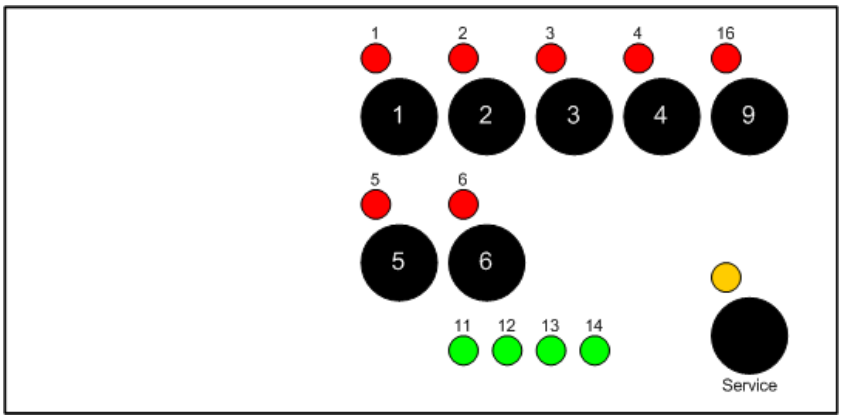


Bild 2.2.2: Kanalnummern der Bedienelemente des LCS310

Visualisierungsbausteine

Mit Visualisierungsbausteinen der Funktionsbibliothek **Visual** können programminterne Daten (Variablen, Parameter) verschiedener Typen (Bit, Word...) auf der IPOCS-Oberfläche beobachtet werden. Dies dient der Unterstützung bei der Applikationsentwicklung (Simulation) sowie der Inbetriebnahme (Online-Modus). Visualisierungsbausteine haben keinen Einfluss auf die eigentliche Applikation im Zielsystem.

Parameter

Parameterbausteine aus der Funktionsbibliothek **Parameter** dienen der Parametrierung von IPOCS-Applikationen. Parameter werden im nichtflüchtigen Speicher des Zielsystems abgelegt und können auch während der Programmabarbeitung online über die Oberfläche von IPOCS-4.1 modifiziert werden.

Netzwerkvariablen (NVs)

Netzwerkvariablen auf Basis des LonTalk-Protokolls werden zur Kommunikation der Geräte untereinander im LonWorks-Netzwerk, verwendet. Über Input-Netzwerkvariablen können Daten aus dem Netzwerk empfangen werden. Ausgangs-Netzwerkvariablen dienen dem Senden von Daten in das Netzwerk. Die Funktionsbibliothek **NV** stellt Funktionsbausteine für die Ein- und Ausgabe von Daten über Netzwerkvariablen zur Verfügung.

Netzwerkvariablen können verschiedene Typen haben. Im Sinne der Interoperabilität sollten nur Standard-Netzwerk-Variablen-Typen (SNVTs) verwendet werden. Diese SNVTs definieren Interpretationsregeln, welche durch spezielle NV-Visualisierungs- und Parameterbausteine unterstützt werden. Sie haben so die Möglichkeit, die Daten in interpretierter Form zu beobachten und einzugeben.

Wie NV´s genau eingesetzt werden, wird in Kapitel 4 näher beschrieben.

Arithmetische Funktionen

Die Funktionsbibliothek **Arithmetik** enthält eine Vielzahl von einfachen und komplexen Funktionsbausteinen für die Verarbeitung von Signalen.

2.3 Schritte der Programmierung

Dieser Abschnitt hilft Ihnen bei der Erstellung der eigentlichen Anwendung.

Aufgabe 2: Programmierung eines Treppenlichtautomaten:

- Anschluss eines Tasters an den Digitaleingang 1
- Anschluss der Leuchte an Relaisausgang 1
- Betätigung des Tasters soll die Leuchte einschalten; nach einer Minute soll die Leuchte automatisch ausschalten

Lösung der Aufgabe 2

Es soll ein Treppenlichtautomat programmiert werden. Ein Digitaleingang-Baustein liest das Signal des angeschlossenen Tasters. Das Eingangssignal wird an einen **MOKY**-Baustein (monostabile Kippstufe) weitergereicht, dessen Ausgangssignal wiederum an einen Relaisausgang-Baustein geleitet wird.

Mit einem Parameterbaustein wird dem Digitaleingang-Baustein und dem Relaisausgang-Baustein die Kanalnummer (1) übergeben. Ein weiterer Parameter definiert die Nachlaufzeit der monostabilen Kippstufe in 1/10 Sekunden (eine Minute entspricht 600). Geben Sie diese Parameter in den entsprechenden Konfigurationsdialogen ein.

Zu Visualisierungszwecken können weiterhin Visualisierungsbausteine (Beobachten mit IPOCS-4.1) und LED-Bausteine (Beobachten am Zielsystem) eingefügt werden.

Einfügen von Funktionsbausteinen

Aktivieren Sie die Ansicht ***Bibliotheken*** des Projektbaumes. Sie sehen alle geladenen Bibliotheken. Per Doppelklick auf einen Bibliotheksnamen werden die Namen aller in dieser Bibliothek befindlichen Bausteine angezeigt. Wählen Sie mit der Maus einen Baustein aus – er wird im Bausteinfenster des Projektbaumes dargestellt. Nun können Sie den Baustein per Drag&Drop-Operation aus dem Bausteinfenster heraus auf die Arbeitsfläche ziehen.

Fügen Sie auf diese Weise folgende Funktionsbausteine in Ihr Arbeitsblatt ein:

Bibliothek	Baustein	Beschreibung
Arithmetik	MOKY	Monostabile Kippstufe
Hardware I/O	DigIn	Digitaleingang
Hardware I/O	DigOut	Relaisausgang
Parameter	ParaWord	Baustein zur Definition eines „Word“- Parameters
Parameter	ParaWord	Baustein zur Definition eines „Word“- Parameters

Tabelle 2.3.1: Einzufügende Funktionsbausteine

Beim Ablegen der Funktionsbausteine auf dem Arbeitsblatt öffnen sich ggf. bausteinspezifische Parametrierdialoge, in denen entsprechende Vorgabewerte eingetragen werden können. Über das lokale Menü (Mauszeiger auf den Funktionsbaustein, rechte Maustaste betätigen) kann der Konfigurationsdialog auch später jederzeit aufgerufen werden, um nachträglich Änderungen an den Parametern vorzunehmen.

Bild 2.3.1: Parametrierfenster eines ParaWord-Bausteins

Wenn Sie mit dem Mauszeiger auf einen Funktionsbaustein im Arbeitsblatt gehen und <STRG>+<F1> drücken, wird die bausteinspezifische Hilfe aufgerufen. Hier finden Sie alle nötigen Informationen zu dem Baustein: Funktion, Anschlüsse, Parameter etc.

Nun haben Sie alle nötigen Bausteine auf der Oberfläche. Diese müssen jetzt noch funktional miteinander verbunden werden, damit Daten zwischen diesen auch ausgetauscht werden können.

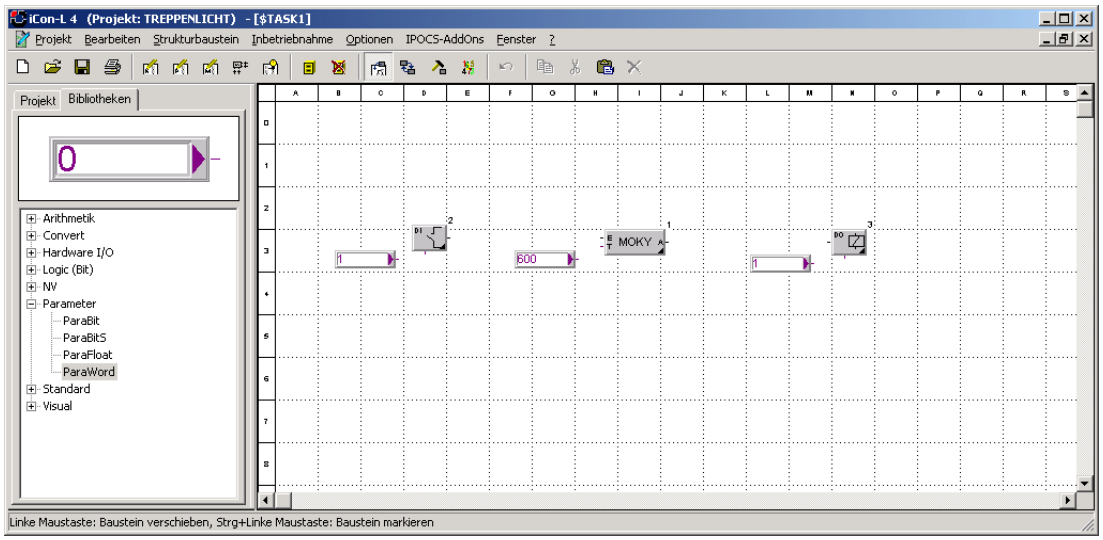


Bild 2.3.2: Bibliotheksansicht im Editiermodus; Einfügen von Funktionsbausteinen

Erstellen von Verbindungen

Funktionsbausteine verfügen über Anschlussstellen, die Ein- und Ausgangsdaten des Bausteins darstellen. Die Anschlüsse sind dem Datentyp entsprechend farblich gekennzeichnet.

Sie erstellen Verbindungen zwischen den Bausteinen wie folgt:

- Klicken Sie mit dem Mauszeiger auf eine Anschlussstelle eines Funktionsbausteins. Der Mauszeiger ändert sein Aussehen von Stift zu einem Fadenkreuz.
- Klicken Sie nun auf den Anschluss eines anderen Bausteins, um die Verbindung zu erstellen.

Hinweis:
Versuchen Sie herauszufinden, welche Verbindungen richtig sind. IPOCS-4.1 verfügt über eine integrierte Typprüfung. Damit wird verhindert, dass Eingänge und Ausgänge mit nicht übereinstimmenden Datentypen verbunden werden. Ebenso werden Kurzschlüsse von Ausgängen verhindert.

Wenn Sie alle Bausteine anhand der Aufgabenstellung richtig miteinander verbunden haben, sollte Ihr Projekt nun so aussehen.

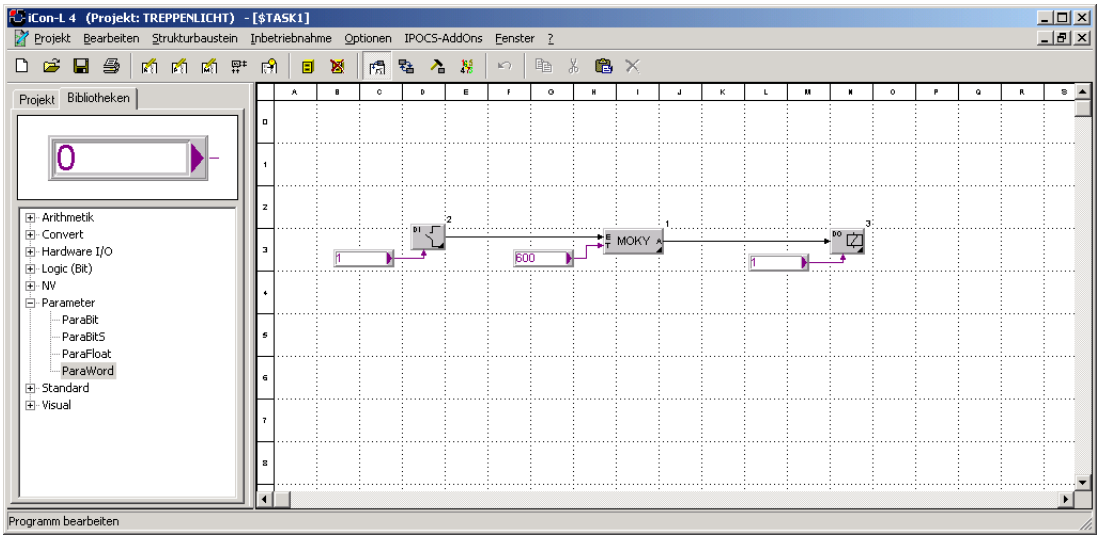


Bild 2.3.3: Projekt mit Bausteinen und definierten Verbindungen

Visualisierungsfunktionen einfügen

Um die Applikation besser beobachten zu können, empfiehlt sich zusätzlich das Einfügen von Visualisierungsbausteinen (**LEDBit** aus der Bibliothek **Visual**) und LED-Bausteinen (**HOut** aus der Bibliothek **Hardware I/O** – beachten Sie die Zuordnung der Kanalnummern) zur optischen Anzeige des Eingangs- und des Ausgangssignals.

Programmbaustein editieren

- Um Bausteine auf dem Arbeitsblatt zu verschieben, klicken Sie den Baustein einfach an und ziehen ihn mit der Maus bis zu seinem Bestimmungsort.
- Wenn Sie Funktionsbausteine von dem Arbeitsblatt löschen wollen, wählen Sie im Menü ***Bearbeiten > Bausteine löschen*** und klicken Sie auf die zu löschenden Bausteine.
- Die Verbindungen können Sie am besten über die lokalen Menüs (Mauszeiger auf die Verbindung, rechte Maustaste betätigen) bearbeiten und löschen.
- Zum Vergrößern / Verkleinern der Ansicht des Arbeitsblattes klicken Sie auf das Arbeitsblatt und betätigen die **+** bzw. **–** Taste des Ziffernblocks der Tastatur.
- Mit Textbausteinen (**Text** aus der Bibliothek **Standard**) kann der Programmbaustein dokumentiert werden.

Bausteinreihenfolge

Jeder Funktionsbaustein, der später vom Zielsystem abgearbeitet werden soll (Ausnahmen: Texte, Visualisierungsbausteine und Parameterbausteine), besitzt eine Bausteinnummer, die fortlaufend vergeben wird und die Abarbeitungsreihenfolge der Funktionsbausteine definiert. Um den Programmablauf effektiv zu gestalten und vor allem um Fehlfunktionen zu vermeiden, sollten Sie die Bausteinnummern entsprechend der gewünschten Abarbeitungsreihenfolge (typisch von Eingängen zu Ausgängen) anpassen. Wählen Sie dazu Menüpunkt ***Bearbeiten > Baustein -Reihenfolge*** und klicken Sie in der gewünschten Reihenfolge auf die Bausteine.

Programmbausteinname

Um die Funktion des programmierten Programmbausteins schon am Namen erkennen zu können, sollte der Programmbausteinname geändert werden. Aktivieren Sie dazu im Projektbaum die Ansicht ***Projekt*** und klicken Sie mit der rechten Maustaste auf den Programmbaustein **TASK1**. Wählen Sie im lokalen Menü ***Umbenennen*** und vergeben Sie einen deskriptiven Namen: **LICHT**.

Im Resultat sollte Ihr Projekt folgenden Aufbau haben:

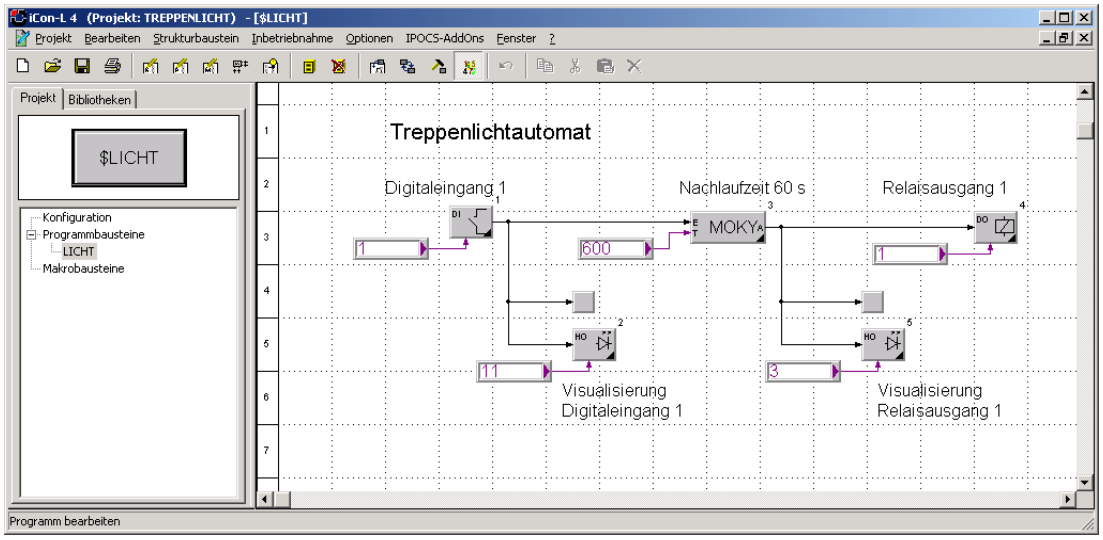


Bild 2.3.4: Programmbaustein *Treppenlichtautomat*

Hinweis:
Jetzt ist ein guter Zeitpunkt, um das erstellte Projekt abzuspeichern. Wählen Sie aus dem Menü ***Projekt > Speichern***.

Wir haben nun einen Programmbaustein erstellt, der eine Treppenlichtautomatikschaltung realisiert.

Sie können das Programm zu einem späteren Zeitpunkt erweitern und verändern. Diese Schritte werden in Kapitel 4 näher erläutert. Zunächst soll aber der Projekttest und der Download des Projektes in ein reales Gerät erläutert werden.

3 Test und Inbetriebnahme

Um das erstellte Programm zu testen, müssen Sie es in ein Zielsystem laden. Eine Besonderheit von IPOCS-4.1 ist die Simulation eines Zielsystems auf dem PC. Damit ergeben sich prinzipiell zwei Testszenarien:

- auf dem PC mit Hilfe einer Geräte-Simulation
- auf einem realen Zielsystem (Gerät) in einem angeschlossenen LonWorks-Netzwerk

In beiden Fällen wechseln Sie als erstes in den Inbetriebnahme-Modus von IPOCS-4.1. Wählen Sie dazu aus dem Menü ***Inbetriebnahme > Eintritt***. Im Projektbaum sehen Sie nun den Aufbau der Applikation (**MAIN**): Es existiert eine Instanz des Programmbausteins **LICHT**.

Hier können Sie die Zykluszeit für diese Instanz des Programmbausteins einstellen. Drücken Sie dazu die rechte Maustaste während der Mauszeiger über dem Programmbaustein platziert ist und wählen Sie aus dem lokalen Menü ***Zykluszeit...*** aus. Geben Sie eine Zykluszeit von 100 (ms) ein. Über die Zykluszeit steuern Sie, wie oft die Instanz des Programmbausteins abgearbeitet wird. Für Temperaturregelungen kann die Zykluszeit viele Sekunden betragen; da wir bei der Treppenlichtsteuerung aber auf Knopfdruck sofort eine Reaktion erwarten, wählen wir eine kleine Zykluszeit.

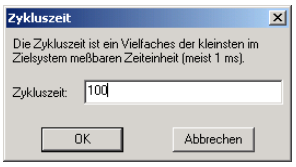


Bild 3.1: Eingabe der Zykluszeit

Hinweis:
Die kleinste Zykluszeit im LCS310 beträgt 100 ms.

Danach müssen Sie in IPOCS-4.1 das Zielsystem anmelden, mit welchem Sie den Test ausführen wollen. Wählen Sie den Menüpunkt ***Inbetriebnahme > Zielsystem anmelden***. Es erscheint ein Dialog, aus dem Sie das gewünschte Zielsystem auswählen und anmelden können.

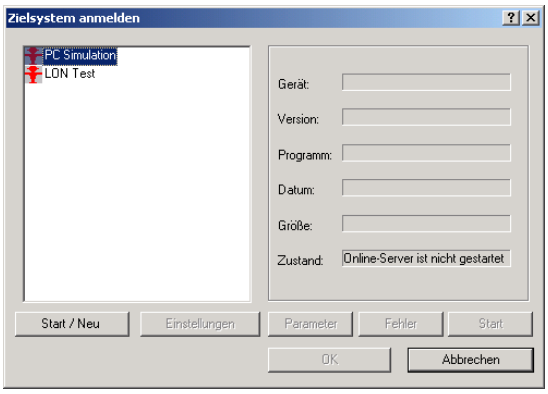


Bild 3.2: Auswahlfenster *Zielsystem anmelden*

3.1 Test mit Hilfe der PC-Simulation

Aufgabe 3: Test der Applikation

- Mit Hilfe der PC-Simulation soll das erstellte Projekt getestet werden.
- Hier soll auch das Simulationsprogramm „Control-Server-Simulator“ zum Einsatz kommen.

Klicken Sie mit der Maus auf das Zielsystem **PC-Simulation** und auf die Taste ***Start/Neu***. Damit ist das Zielsystem angemeldet. Als nächstes soll das Projekt in das Zielsystem geladen werden. Wählen Sie dazu aus dem Menü ***Inbetriebnahme > Anwenderprogramm laden***. Im ***Messages***-Fenster werden Arbeitsschritte sowie auftretende Fehler ausgegeben.

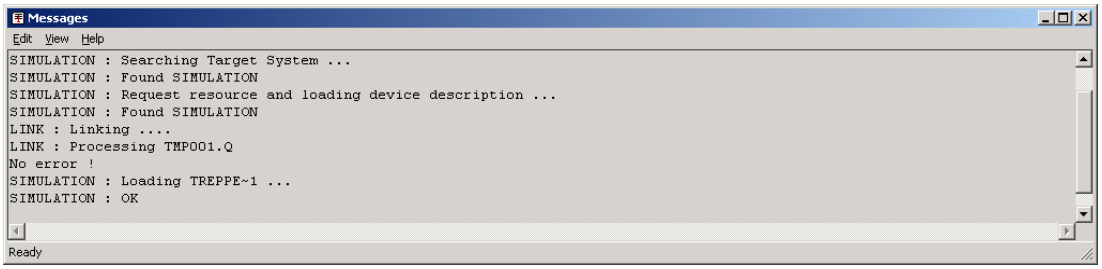


Bild 3.1.1: *Messages*-Fenster

Wenn keine Fehler gefunden wurden, geht IPOCS-4.1 nach dem Laden in den Online-Modus. Navigieren Sie im Projektbaum zur Instanz des Programmbausteins, um die Treppenlichtsteuerung beobachten zu können.

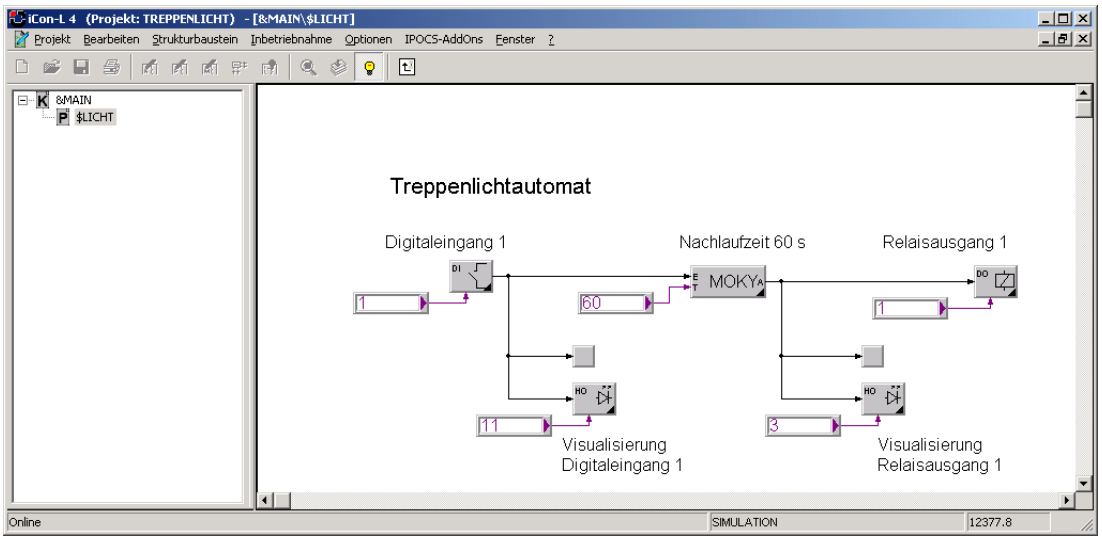


Bild 3.1.2: geöffnetes Projekt während der Simulationsphase

Um speziell die peripheren Einheiten des LON-Control-Servers anschaulich darzustellen, kann das Simulationstool **ControlServer Simulator** über den Menüpunkt ***IPOCS-AddOns > ControlServer Simulator*** gestartet werden. Es erscheint ein Dialog, in dem alle Hardwareein- und -ausgänge des LON-Control-Servers visualisiert werden. Damit ist es möglich, während der Simulationsphase die Hardwarefunktionalitäten nachzubilden.

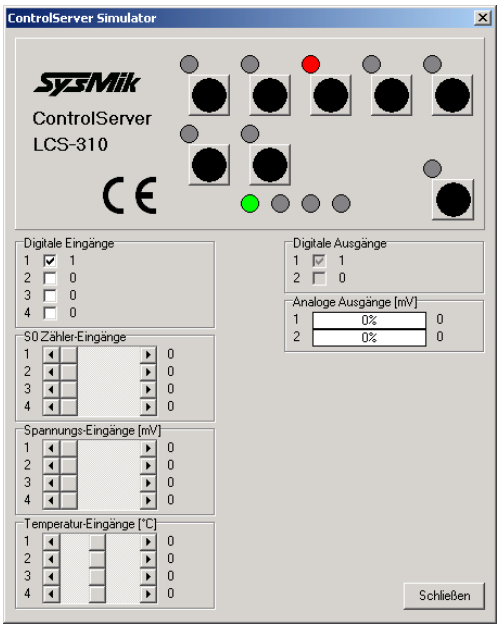


Bild 3.1.3: Simulationsprogramm für LCS310

Wenn Sie durch einen Klick auf den Digitalen Eingang 1 das Drücken des Treppenlichttasters simulieren, können Sie beobachten, wie der Digitale Ausgang 1 eingeschaltet wird. Weiterhin sehen Sie die programmierten LEDs des Zielsystems im ControlServer Simulator leuchten.

Parallel dazu können Sie in IPOCS-4.1 anhand der Visualisierungsbausteine den Zustand der Signale des Treppenlichtautomaten beobachten.

Weiterhin können Sie sich durch einen Mausklick auf eine der Verbindungen den jeweiligen aktuellen Signalwert anzeigen lassen.

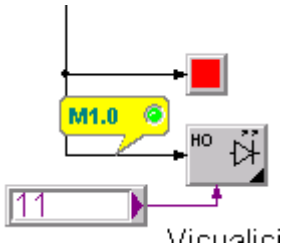


Bild 3.1.4: Fenster zum Anzeigen eines Signalwertes

Parameter lassen sich online ändern. So können Sie zum Beispiel die Nachlaufzeit auf 30 Sekunden setzten. Klicken Sie dazu einfach auf den Parameterbaustein und geben Sie im sich öffnenden Konfigurationsdialog die neue Nachlaufzeit (300) ein.

3.2 Download in ein reales Zielsystem

Neben der Simulation auf dem PC können Sie das Projekt auch direkt in ein IPOCS-programmierbares Gerät laden und dort testen bzw. damit arbeiten. In unserem Beispiel verwenden wir den **LON-Control-Server** LCS310.

Hierzu wählen Sie im Fenster ***Zielsystem*** das System **LON Test**.

Bild 3.2.1: LON-Control-Server LCS310

Hinweis:
Das Zielsystem **LON-Test** sollte ausschließlich für reine Tests verwendet werden. Erstellen Sie für Projekte eigene Kommunikationstemplates mit unikaten und deskriptiven Namen.

Nachdem Sie auf die Schaltfläche ***Start/Neu*** gedrückt haben sucht IPOCS-4.1 das Zielsystem. Da aber weder der Kommunikationskanal noch die Adresse des Zielsystems bekannt sind, wird Ihnen im Fenster ***Messages*** die Fehlermeldung **Communication Error** angezeigt.

Hinweis:
Wenn ein reales Zielsystem verwendet wird, muss vorher sichergestellt sein, dass eine physikalische Verbindung zum LonWorks-Netzwerk besteht (Interfacemodul) und ggf. entsprechende Zusatzprogramme (z.B. SLTA-Link-Manager) gestartet sind.

Durch Klicken auf die Schaltfläche ***Einstellungen*** gelangen Sie in das Fenster ***Einstellungen***.

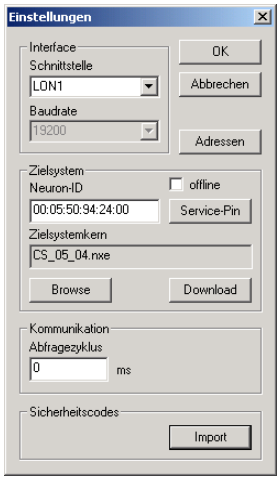


Bild 3.2.2: Dialog *Einstellungen*

Hier werden folgende Einstellungen vorgenommen:

- Festlegen des Kommunikationskanals zwischen PC und Zielsystem Bei Verwendung des IFM-RS232 als Interfacemodul tragen Sie hier die verwendete serielle Schnittstelle sowie die konfigurierte Baudrate ein.
- Definition des Zielsystems über dessen Neuron-ID. Neben dem manuellen Eintragen der Neuron-ID können Sie auch die Schaltfläche ***Service-Pin*** betätigen und dann am Zielsystem den Service-Taster drücken. Dabei wird die Neuron-ID empfangen und automatisch als Zielsystemadresse übernommen.
- Es kann eine APB- bzw. NXE-Datei (z.B. ein neuer Zielsystemkern) in das Zielsystem geladen werden: Klicken Sie auf ***Browse***, um eine Datei auszuwählen und dann ***Download***.
- Speziell für die Erstellung von APB- und NXE-Dateien im Büro (offline) dient das Auswahlfeld ***Offline***. Ist dies aktiviert, wird ein Zielsystem mit dem ausgewählten Zielsystemkern emuliert, so dass ein Applikationsdownload der IPOCS-Applikation und die damit verbundene Erzeugung der Programmierdateien auch ohne reales Zielsystem ausgelöst werden kann.
- Festlegen des minimalen Abfragezyklus des Zielsystems zur Drosselung der Netzwerkbelastung im Online-Modus von IPOCS-4.1.

Damit eine bidirektionale Kommunikation zwischen PC und Zielsystem stattfinden kann, müssen die Adresseinstellungen von Interfacemodul und Zielsystem aufeinander abgestimmt werden. Um die Adressen anzupassen, klicken Sie mit der Maus auf die Schaltfläche ***Adressen***.

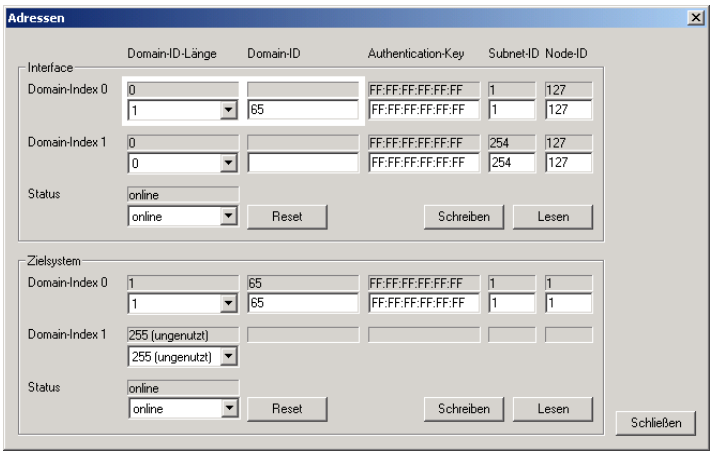


Bild 3.2.3: Dialog *Adressen*

Daraufhin öffnet sich ein Fenster, in welchem die spezifischen Werte für Domain-ID, Subnet-ID, Node-ID für das Netzwerk-Interface und das zu verwendende Zielsystem angezeigt werden. Wenn einzelne Elemente der Adresseinstellungen von Interface und Zielsystem nicht zueinander passen, schlägt IPOCS eine entsprechende Änderung vor und markiert diese mit einem weißen Rahmen. Man kann hier zwischen dem Vorschlag des Systems oder eigenen Vorgaben wählen. Weiterhin kann hier auch der Status für das Interface sowie das Zielsystem festgelegt werden. Nach Abschluss der Adresseinstellung müssen diese über die jeweilige Schaltfläche ***Schreiben*** in das Gerät geschrieben werden. Über die Schaltfläche ***Reset*** können Sie einen Software-Reset des Gerätes durchführen.

Nach Abschluss der Einstellungen sollte das Zielsystem erkannt werden. Nach dem Laden der Applikation können Sie den Treppenlichtautomaten direkt am Gerät testen. Auch hier können Sie im Online-Modus von IPOCS-4.1 auf die Signale und Parameter der IPOCS-Applikation zugreifen.

3.3 Schritte der Inbetriebnahme des realen Zielsystems

Um die in Kapitel 3.2 erläuterte Vorgehensweise zu festigen, soll diese nun noch einmal in Checklistenform anschaulich dargestellt werden.

Checkliste 1: Zielsystem anmelden

1. Sie haben Ihr Projekt erstellt (entsprechend den Arbeitsschritten der vorangegangenen Kapitel) und gespeichert.
2. Sie verfügen über ein Zielsystem und ein Interfacemodul.
3. Wählen Sie aus dem Menü ***Inbetriebnahme > Anwenderprogramm laden***.
4. Danach klicken Sie im Fenster ***Zielsystem anmelden*** mit der rechten Maustaste auf die Liste der Zielsysteme und wählen aus dem lokalen Menü den Eintrag ***Template hinzufügen***.
5. Geben Sie dem neuen Kommunikationstemplate einen unikaten Namen sowie eine deskriptive Beschreibung Ihres Zielsystems. Wählen Sie ***Serverklasse*** **IO32NS** und ***Kommunikationstreiber*** **iconlon.dll** aus. Bestätigen Sie die Angaben mit ***OK***.
6. Markieren Sie das neue Zielsystem und melden Sie es mit Klicken auf die Schaltfläche ***Start/Neu*** an.
7. Weil das Interface und das Zielsystem noch nicht bekannt sind, wird im Message-Fenster die Meldung **Communication Error!** ausgegeben (die Schaltfläche ***OK*** im Fenster ***Zielsystem anmelden*** ist deaktiviert).

Checkliste 2: Interfacemodul konfigurieren

1. Zuerst müssen Sie das Interface definieren und konfigurieren, indem Sie im Dialog ***Einstellungen*** im Bereich ***Interface*** den logischen Interfacenamen oder bei Nutzung des IFM-RS232 die serielle Schnittstelle sowie die Baudrate festlegen.
2. Zur Überprüfung der Adress-Einstellungen klicken Sie auf die Schaltfläche ***Adressen***.
3. In diesem Dialog können Sie die Adresseinstellungen des Interfaces angezeigt und verändert werden. Dabei gilt es, folgende Regel zu beachten:
   - Das Interface muss in der Domain der Länge 0 adressiert sein (unikate Subnet / Node – Adresse), damit Service-Nachrichten empfangen werden können.
4. Das Programm zeigt Ihnen die aktuell gesetzten Einstellungen an (in grauer Fläche, ggf. Anzeige von **Lesefehler**, wenn die Informationen nicht gelesen werden können) und erarbeitet daraus einen Vorschlag für die günstigsten Einstellungen (korrekte Domain-ID, Subnet-ID, Node-ID sowie Status auf **online**). Sie können diese Einstellungen beibehalten oder nach projektspezifischen Gesichtspunkten ändern.
5. Durch Klicken auf die Schaltfläche ***Schreiben*** werden die Änderungen bestätigt und übernommen.
6. Um den Dialog ***Einstellungen*** zu beenden, klicken Sie auf die Schaltfläche ***Schließen***.

Checkliste 3: Zielsystem konfigurieren

1. Danach muss das Zielsystem definiert und konfiguriert werden. Dazu geben Sie im Bereich ***Zielsystem*** die Neuron-ID des Zielsystems an. Diese kann manuell bzw. über den Empfang der Service-Nachricht (Klicken Sie auf die Schaltfläche ***Service-Pin*** und drücken Sie den Service-Taster auf dem Zielsystem) erfolgen.
2. Zur Überprüfung der Adress-Einstellungen des Zielsystems klicken Sie erneut auf die Schaltfläche ***Adressen***, (entsprechend Checkliste 2)
3. Dabei gilt es folgende Regel zu beachten:
   - Interface und Zielsystem müssen in der gleichen Domain (Index 0) liegen und eine unikate Subnet / Node – Adresse besitzen.
4. Das Programm zeigt Ihnen die zur Zeit gesetzten Einstellungen an (in grauer Fläche) und erarbeitet daraus einen Vorschlag für die günstigsten Einstellungen (korrekte Domain-ID, Subnet-ID, Node-ID sowie Status auf **online**). Sie können diese Einstellungen beibehalten oder nach projektspezifischen Gesichtspunkten ändern.
5. Durch Klicken auf die Schaltfläche ***Schreiben*** werden die Änderungen bestätigt und übernommen.
6. Um den Dialog ***Einstellungen*** zu beenden, klicken Sie auf die Schaltfläche ***Schließen***.

Danach versucht der Computer erneut Kontakt mit dem Zielsystem aufzunehmen (Schaltfläche ***OK*** im Fenster ***Zielsystem*** ist aktiviert).

Hinweis:
Bei Kommunikationsschwierigkeiten mit dem Interfacemodul oder Zielsystem können Sie durch Klicken auf die Schaltfläche ***Reset*** im Dialog ***Adressen*** einen Software-Reset durchführen.

4 Netzwerkfunktionen

Nachdem Sie das Zielsystem als eigenständige Einheit programmiert haben, sollen Sie nun die Netzwerkfähigkeiten der LonWorks-Zielsysteme kennenlernen. Die Netzwerkkommunikation der LonWorks-Geräte wird über Netzwerkvariablen (NVs) realisiert. Geräte verfügen über Eingangs-Netzwerkvariablen und Ausgangs-Netzwerkvariablen. Im Rahmen der Systemintegration werden die Netzwerkvariablen mehrerer Geräte entsprechend der zu realisierenden Funktion miteinander verbunden.

Hinweis:
Mit IPOCS Version 4.1.12 wurde ein neuer **Lon Configurator** mit verbesserter Funktionalität eingeführt, der über das Menü ***IPOCS- AddOns > Lon Configurator*** zu starten ist. Zur Fortsetzung der Übung mit diesem Lon Configurator lesen Sie bitte das „Dokument Erste Schritte mit dem Lon Configurator“. Dieses Dokument ersetzt die folgenden Kapitel 4.1 und 4.2. Ältere Bestandsprojekte bieten diesen neuen Lon Configurator nicht an; darum bleiben die erwähnten Kapitel erhalten.
Indem man bei der Installation einer neuen IPOCS-Version das Auswahlfeld ***Projekte übernehmen*** aktiviert, werden die Projekte aus dem Projektverzeichnis einer älteren IPOCS-Installation in das Projektverzeichnis der neuen IPOCS-Installation kopiert und dort an die neue IPOCS-Version angepasst. Dadurch wird auch der neue Lon Configurator verfügbar. Die Originalprojekte bleiben dabei selbstverständlich unverändert.

4.1 Erweiterung des Projektes um Netzwerkvariablen

Aufgabe 4: Ändern der in Kapitel 2.3 erstellten Applikation:

- Alternative Ansteuerung des Treppenlichtautomaten über eine Eingangs-Netzwerkvariable.
- Ausgabe des Ausgangssignals über eine Ausgangs-Netzwerkvariable.

Im ersten Schritt kehren Sie vom Online-Modus in den Programmiermodus zurück. Dazu wählen Sie zunächst den Menüpunkt ***Inbetriebnahme > Offline*** danach den Menüpunkt ***Bearbeiten > Programmbaustein*** und wählen den Programmbaustein **LICHT** aus.

Jetzt können Sie mit dem schon erworbenen Wissen Bausteine dem Projekt hinzufügen und Verbindungen löschen und neu erstellen.

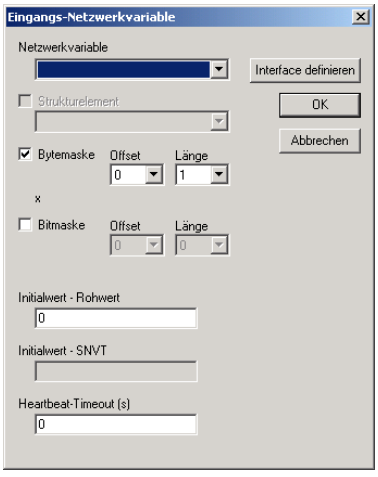


Bild 4.1.1: Konfigurationsdialog *Eingangs-Netzwerkvariable*

Sie benötigen jeweils einen Eingangs-Netzwerkvariablen-Baustein **NVIW02** und einen Ausgangs-Netzwerkvariablen-Baustein **NVOW02** aus der Bibliothek **NV**.

Beim Anlegen des Funktionsbausteins **NVIW02** erscheint folgender Konfigurationsdialog.

Da noch keine Netzwerkvariablen für das Programm definiert wurden, können Sie auch noch keine im Auswahlfeld ***Netzwerkvariable*** auswählen. Klicken Sie also auf ***Interface definieren***, um das Netzwerkvariableninterface der Applikation zu bestimmen.

In dem sich daraufhin öffnenden Dialog Netzwerk Interface haben Sie die Möglichkeit,

- die Programm-ID der Applikation einzugeben (***Programm-ID***),
- in den LonMark-Modus wechseln (***LonMark***) – siehe Kapitel 4.1
- den Dokumentationsstring zu definieren (***Knoten-Selbst-Dokumentation***),
- die Firmwareversion des Zielsystems einzutragen (***Firmwareversion***),
- NVs neu anzulegen (***Neu***),
- die Eigenschaften von NVs zu ändern (***Eigenschaften***),
- NVs zu löschen (***Löschen***),
- die Reihenfolge der NVs zu verändern (***Nach oben*** / ***Nach unten***),
- NVs aus einem anderen Projekt zu importieren (***Import***) und
- das aktuelle NV-Interface zu exportieren (***Export***).

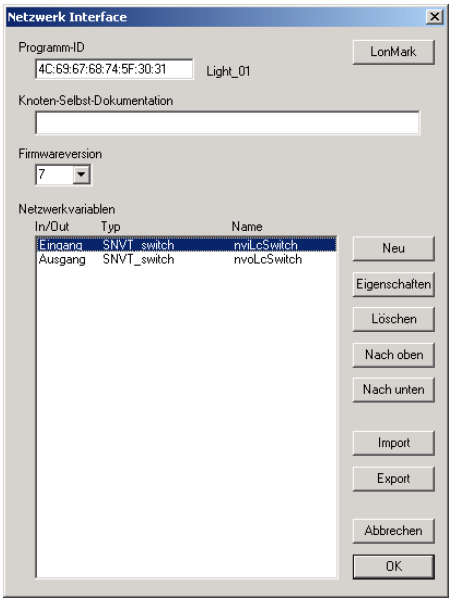


Bild 4.1.2: Dialog *Netzwerk Interface*

Durch einen Mausklick auf die Schaltfläche ***Neu*** öffnet sich der Dialog ***Netzwerkvariable Eigenschaften***.

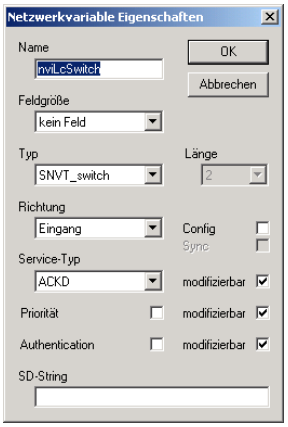


Bild 4.1.3: Dialog *Netzwerkvariable Eigenschaften*

In diesem Dialog können Sie neue NVs erstellen, indem Sie

- einen Namen für die NV eingeben,
- die Feldgröße der NV bestimmen, d.h. wie viele NVs dieses Variablentyps sollen als Feld erstellt werden,
- den Variablentyp (***Typ***) angeben; wenn Sie hier **UNVT** (nutzerdefinierter Typ) auswählen, dann können Sie im Feld ***Länge*** die Länge der Variable angeben.
- festlegen ob es sich um eine Eingangs-NV oder Ausgangs-NV handelt,
- den Service-Typ festlegen.

Legen Sie nun die folgenden Netzwerkvariablen an:

Name	nviLcSwitch	nvoLcSwitch
Feldgröße	kein Feld	kein Feld
Typ	SNVT_switch	SNVT_switch
Richtung	Eingang	Ausgang
Service-Typ	ACKD	ACKD

Tabelle 4.1.1: Definition der Netzwerk-Variablen

Schließen Sie den Dialog ***Netzwerk Interface*** mit ***OK***. Dabei wird automatisch die Interfacedatei (*.xif) sowie eine Texttabelle der definierten Netzwerkvariablen (*.csv – zu öffnen mit MS Excel) im Projektverzeichnis angelegt.

Jetzt können Sie dem NV-Baustein im Dialog ***Eingang Netzwerkvariable*** die Eingangs-NV zuweisen. Der NV-Baustein soll nur den Wert des Strukturelements **state** der Variable **nviLcSwitch** vom Typ **SNVT_switch** an die Applikation zur Auswertung weiterleiten. Markieren Sie also dieses ***Strukturelement***. Den Initialwert (Wert bis zum ersten Empfang eines Netzwerkvariablen-Updates über das LonWorks-Netzwerk) belassen Sie auf **0**.

Im Eingabefeld ***Heartbeat-Timeout (s)*** können Sie festlegen, in welchem Zeitabstand eine Aktualisierung der Netzwerkvariable erfolgen muss. Geschieht diese Aktualisierung nicht in der eingestellten Zeit, wird der Heartbeat-Ausgabewert des Funktionsbausteins von **ON** auf **OFF** gesetzt. Diese Heartbeat-Überwachung der Netzwerkvariable bleibt deaktiviert (**0**).

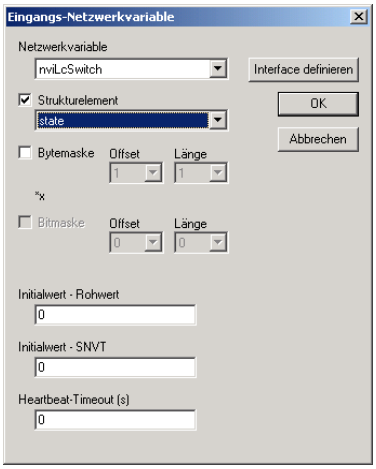


Bild 4.1.4: Dialog *Eingangs-Netzwerkvariable* mit Einstellungen

Um das Signal der Eingangs-Netzwerkvariable (Word) mit dem Signal des Digitalen Einganges (Bit) zu verbinden, benötigen Sie einen Typkonvertierungsbaustein **W2B**

aus der Bibliothek **Convert** sowie eine Oder-Verknüpfung (Baustein **OR** aus der Bibliothek **Arithmetik**).

Fügen Sie nun einen Ausgangs-Netzwerkvaribalen-Baustein **NVOW02** in den Programmbaustein ein. Im Konfigurationsdialog ***Ausgangs-Netzwerkvariable*** wählen Sie die **nvoLcSwitch** aus und deaktivieren die Optionen ***Strukturelement*** und ***Bytemaske***, da die Werte beider Strukturelemente der Netzwerkvariable von Typ **SNVT_switch** beschrieben werden sollen: 100%:1 (**ON**) bzw. 0%:0 (**OFF**).

Da von der Treppenlichtautomatik ausschließlich digitale Schaltsignale gebildet werden, muss jede Änderung sofort übertragen werden. ***Kein NV-Update auslösen*** bleibt deaktiviert; Die Einstellungen von ***MinSendTime***, ***MaxSendTime*** und ***Minimale übertragene Änderung*** (SendOnDelta) bleiben auf **0**.

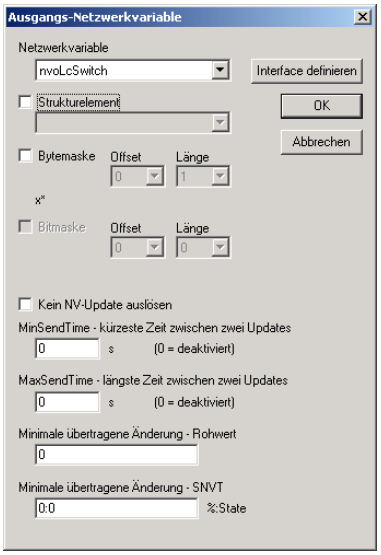


Bild 4.1.5: Konfigurationsdialog *Ausgangs-Netzwerkvariable*

Um das Schaltsignal (Bit) entsprechend der Typdefinition der SNVT_switch auszugeben, verwenden Sie zwei SNVT-Parameter (**ParaWord02** aus Bibliothek **NV**), die Sie im Konfigurationsdialog auf die Werte ON (100%:1) bzw. OFF (0%:0) stellen.

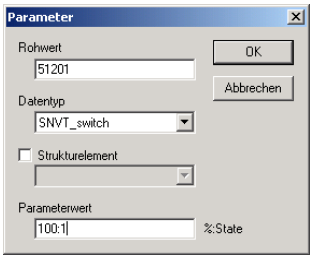


Bild 4.1.6: Konfigurationsdialog der SNVT-Parameter

Über ein Auswahltor (Baustein **AWT** der Bibliothek **Arithmetik**) stellen Sie die Verknüpfung zur Ausgangs-Netzwerkvariable her. Nach erneuter Festlegung der Bausteinreihenfolge sollte Ihr Projekt nun folgenden Aufbau haben.

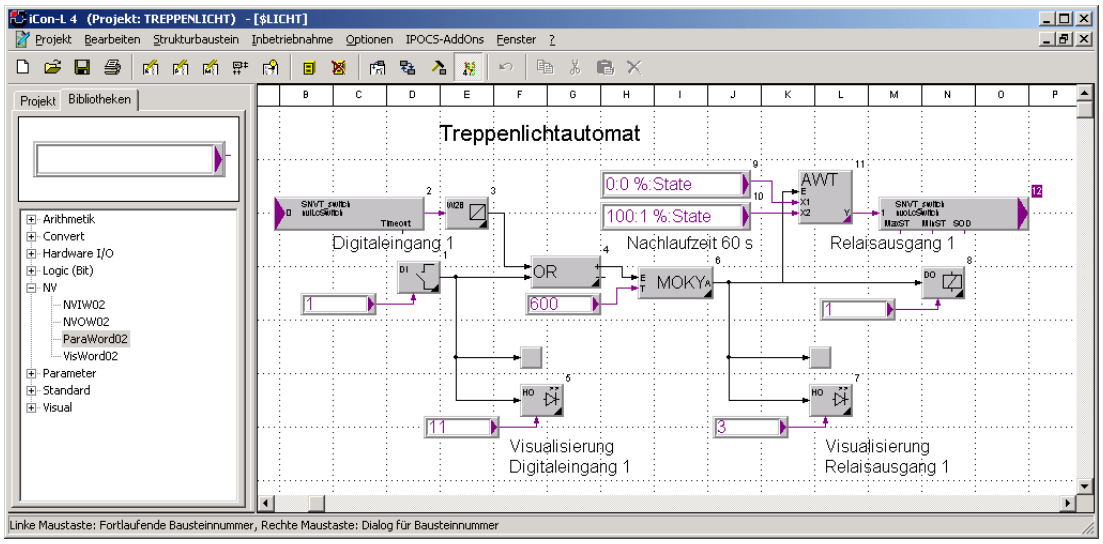


Bild 4.1.7: Das geänderte Projekt mit neuen Bausteinen

Das Projekt ist jetzt den geänderten Erfordernissen angepasst und kann erneut getestet werden. Hierzu starten Sie wie in Kapitel 3.1 beschrieben entweder die PC-Simulation, oder Sie laden das Projekt in das reale Zielsystem (Kapitel 3.2).

In der PC-Simulation können Updates von Eingangs-Netzwerkvariablen simuliert werden, indem auf den **NVIW**-Baustein geklickt wird und im Konfigurationsdialog ein neuer Initialwert vereinbart wird.

4.2 LonMark-Unterstützung

LonMark-konforme Applikationen zeichnen sich durch eine standardisierte objektorientierte Dokumentation des Netzwerkinterfaces sowie durch eine nach spezifizierten Regeln zusammengestellte Programm-ID aus.

Vorteile von LonMark-konformen Applikationen bestehen vor allem hinsichtlich der Integrationsfähigkeit in LonWorks-Netzwerke sowie in der Handhabung der programmierten Geräte mit Netzwerkmanagementtools.

Ab IPOCS4.1 erhalten Sie aktive Unterstützung bei der Erstellung LonMark-konformer Applikationen.

Beim Anlegen des Netzwerkvariableninterfaces im Dialog ***Netzwerk Interface*** (Kapitel 4.1) können Sie über die Schaltfläche ***LonMark*** einen Dialog öffnen, der die auf Basis der Selbstdokumentationstexte definierten LonMark-Objekte anzeigt.

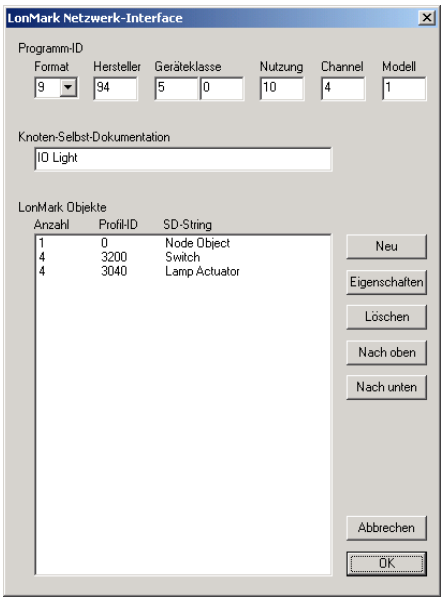


Bild 4.2.1: Dialog *LonMark Netzwerk-Interface*

Im oberen Teil des Dialoges ***LonMark Netzwerk-Interface*** wird die ***Programm-ID*** in die nach LonMark definierten Bestandteile zerlegt angezeigt:

- ***Format***: **8** für zertifizierte Produkte / **9** für alle anderen Geräte – also auch für IPOCS-Applikationen
- ***Hersteller***: Hier wird der von der LonMark-Organisation vergebene Herstellercode Ihres Unternehmens eingetragen.
- ***Geräteklasse***: Hier wird die primäre Funktion des Gerätes in zwei Stellen (Klasse.Unterklasse) angegeben. Es gibt eine Vielzahl von Codes, die von der LonMark-Organisation definiert werden. Einige sind:
  - I/O (**5.0**)
  - Generische Mensch-Maschine-Schnittstelle (**8.0**)
  - Licht (**30.0**)
  - HVAC (**80.0**)
  - Industrie (**130.0**)
- ***Nutzung***: Hier wird der Verwendungszweck definiert. Es gibt eine Vielzahl von Codes, die von der LonMark-Organisation definiert werden. Einige sind:
  - Industrie (**2**)
  - Kommerziell (**3**)
  - Wohnen (**5**)
  - Allgemein (**10**)
- ***Channel***: Dieser Bereich der Programm-ID enthält einen Code für die verwendete Übertragungstechnik / Transceivertechnologie. Das Medium TP/FT-10 hat den Code **4**.
- ***Modell***: Dieses Feld dient der Unterscheidung verschiedener Applikationsversionen und wird vom Applikationsprogrammierer frei vergeben.

Die vollständigen Informationen erhalten Sie in den „LonMark Application-Layer Interoperability Guidelines“, die Sie von der Homepage der LonMark-Organisation www.lonmark.org laden können.

Sie können einen Dokumentationstext eingeben, der Ihre Applikation beschreibt. Die LonMark-typischen Elemente werden später bei Verlassen dieses Dialoges automatisch zu diesem Text hinzugefügt.

Sie sehen eine Liste mit den LonMark-Objekten, die in Ihrer IPOCS-Applikation vorhanden sind. Über die Schaltflächen können Sie neue LonMark-Objekte erstellen, bestehende LonMark-Objekte ändern, löschen und umsortieren.

Über die Schaltflächen ***Neu*** oder ***Eigenschaften*** öffnen Sie den Dialog ***LonMark Objekt Eigenschaften***, der alle Informationen über das LonMark-Objekt anzeigt.

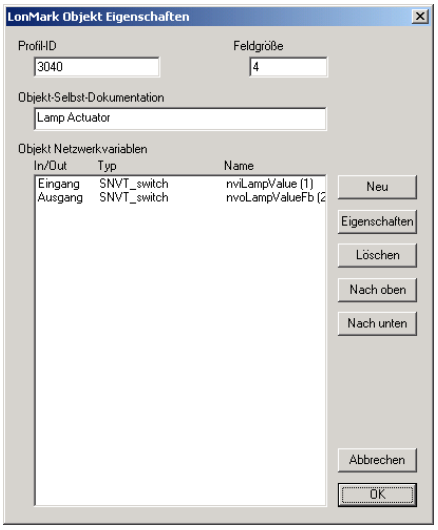


Bild 4.2.2: Dialog *LonMark Objekt Eigenschaften*

Die ***Profil-ID*** bezeichnet das LonMark-Profil, welches dem LonMark-Objekt zugrunde liegt. Die LonMark-Organisation hat eine Vielzahl von Profilen bereits standardisiert; die LonMark-Mitglieder erarbeiten ständig weitere Profile und reichen diese Vorschläge bei LonMark ein.

Es gibt darüber hinaus eine Reihe generischer LonMark-Objekte, die sich für die Applikationsprogrammierung mit IPOCS anbieten:

- Open Loop Sensor Object (**1**)
- Closed Loop Sensor Object (**2**)
- Open Loop Actuator Object (**3**)
- Closed Loop Actuator Object (**4**)
- Controller Object (**5**)

Wenn Sie ein LonMark-Objekt nach einem Profil implementieren wollen, dann sollten Sie die entsprechende Dokumentation zu Hilfe nehmen. Diese erhalten Sie auf der Homepage der LonMark-Organisation www.lonmark.org unter „LonMark Functional Profiles“.

Die ***Feldgröße*** legt fest, wie viele Instanzen des LonMark-Objektes Sie implementieren wollen. Damit lassen sich z.B. sehr einfach mehrere gleiche Sensorobjekte für mehrere Eingangskanäle realisieren.

Sie können dam LonMark-Objekt über die ***Objekt-Selbst-Dokumentation*** eine Bezeichnung geben.

In der Liste ***Objekt Netzwerkvariablen*** werden alle Netzwerkvariablen des LonMark-Objektes angezeigt, die über die Schaltflächen erzeugt, modifiziert, gelöscht und umsortiert werden können.

Über die Schaltflächen ***Neu*** oder ***Eigenschaften*** öffnen Sie den Dialog ***LonMark NV Eigenschaften***, der alle Informationen über die Netzwerkvariable anzeigt (vgl. Kapitel 4.1).

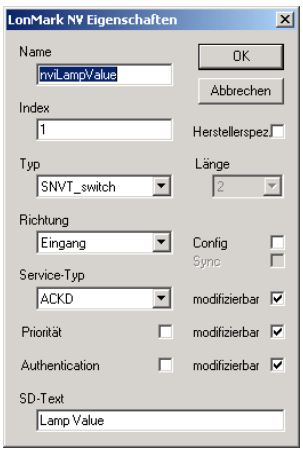


Bild 4.2.3: Dialog *LonMark NV Eigenschaften*

Der Großteil der Informationen kann aus den Profildefinitionen entnommen werden. Der ***Index*** definiert den Bezug zur Profildefinition. Alle im LonMark-Profil enthaltenen Netzwerkvariablen haben einen eindeutigen Index. Wollen Sie darüber hinaus weitere herstellerspezifische (bzw. applikationsspezifische) Netzwerkvariablen anlegen, markieren Sie das Auswahlfeld ***Herstellerspez.*** und geben Sie einen unikaten ***Index*** an.

Über die Aktivierung des Auswahlfeldes ***Config*** definieren Sie eine Configuration Property in Form einer Konfigurationsnetzwerkvariablen. In diesem Fall ist bei ***Index*** die entsprechende SCPT-Nummer einzugeben. Wollen Sie darüber hinaus weitere herstellerspezifische (bzw. applikationsspezifische) Configuration Properties anlegen, markieren Sie das Auswahlfeld ***Herstellerspez.*** und geben Sie einen unikaten ***Index*** an.

Wenn alle Informationen zu den LonMark-Objekten und Ihren Netzwerkvariablen eingegeben sind, verlassen Sie den LonMark-Modus, indem Sie den Dialog ***LonMark Netzwerk-Interface*** über ***OK*** verlassen. Alle LonMark-spezifischen Informationen werden nun in den Selbstdokumentationstexten eingebaut und das flache Netzwerkvariableninterface der Applikation erzeugt und angezeigt.

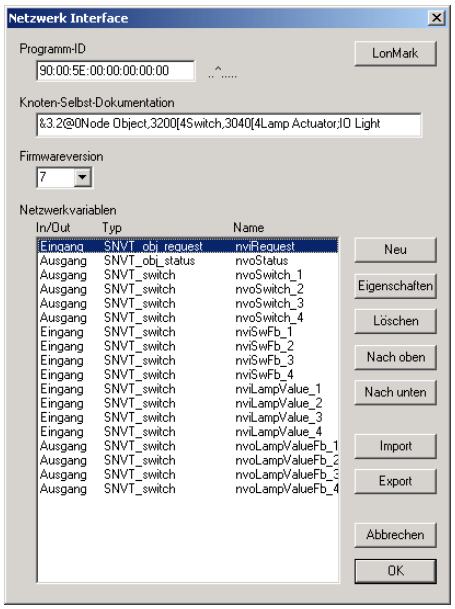


Bild 4.2.4: Dialog *Netzwerk Interface* nach Verlassen des LonMark-Modus

Checkliste 4: LonMark-konformes Netzwerkinterface erstellen

1. Laden Sie von der Homepage der LonMark-Organisation www.lonmark.org die aktuelle Fassung der „LonMark Application-Layer Interoperability Guidelines“ sowie die Profildefinitionen, auf deren Basis Sie LonMark-Objekte erzeugen wollen.
2. Legen Sie in Ihrer Applikation einen Netzwerkvariablenbaustein an und betätigen Sie die Schaltfläche ***Interface definieren*** im Konfigurationsdialog.
3. Betätigen Sie die Schaltfläche ***LonMark*** im Dialog ***Netzwerk Interface***
4. Geben Sie Werte für die Elemente der ***Programm-ID*** sowie die ***Knoten-Selbst-Dokumentation*** ein.
5. Legen Sie mit ***Neu*** ein neues LonMark-Objekt an; Geben Sie die ***Profil-ID*** (siehe Profildokumentation), die Anzahl der Objekte, die Sie anlegen möchten sowie ggf. die ***Objekt-Selbst-Dokumentation*** ein.
6. Erzeugen Sie mit ***Neu*** eine neue Netzwerkvariable in dem LonMark-Objekt; Geben Sie entsprechend der Profildokumentation ***Name***, ***Index***, ***Typ***, ***Richtung*** und ggf. einen Selbstdokumentationstext (***SD-Text***) ein; bestätigen Sie mit ***OK***.
7. Erzeugen Sie mit ***Neu*** einen neue Configuration Property in dem LonMark-Objekt; Aktivieren Sie ***Config***; Geben Sie entsprechend der Profildokumentation ***Name***, SCPT-Nummer (***Index***), ***Typ***, ***Richtung*** (**Input**) und ggf. einen Selbstdokumentationstext (***SD-Text***) ein; bestätigen Sie mit ***OK***.
8. Wiederholen Sie die Schritte 6 und 7, bis das LonMark-Objekt vollständig definiert ist.
9. Wiederholen Sie die Schritte 5 bis 8, bis Sie alle LonMark-Objekte definiert haben; verlassen Sie alle Dialoge mit ***OK.***

4.3 Systemintegration

Hinweis:
Wenn das Gerät mit einem Netzwerkmanagementtool (NMT) installiert werden soll, muss vorher das mit IPOCS-4.1 erstellte Projekt in das Zielsystem geladen worden sein.

Ein IPOCS-programmierbares Gerät mit Netzwerkvariablen kann als eigenständiges Gerät betrieben werden. Den vollen Funktionsumfang entfaltet es aber erst durch die Verbindung der Netzwerkvariablen mit Netzwerkvariablen anderer Geräte in einem LonWorks-Netzwerk. Die Erstellung dieser Verbindungen ist typische Aufgabe der Systemintegration.

Um IPOCS-4.1 programmierte Geräte mit Hilfe von Netzwerk-Management-Tools in Projekten integrieren zu können, sind folgende Arbeitsgänge nötig.

- Erstellen der anwendungsspezifischen Applikation in IPOCS-4.1.
- Dabei: Automatische Generierung einer *.xif-Datei durch IPOCS-4.1 im spezifischen Projektverzeichnis (Bsp.: ../ipocs4/project/treppenlicht) unter dem jeweiligen Projektnamen mit der Endung *.xif (Bsp.: treppenlicht.xif).
- Testen der Applikation mit Hilfe der PC-Simulation.
- Download der Applikation in das Zielsystem im LonWorks-Netzwerk.
- Importieren der generierten *.xif-Datei in beliebige Netzwerkmanagementtools. Dabei müssen die Vorgehenshinweise der einzelnen Tools beachtet werden, um auf Grundlage der von IPOCS-4.1 generierten *.xif-Dateien die spezifischen Geräte anzulegen.
- Im Netzwerkmanagementtool können die Netzwerkvariablen der Geräte untereinander verbunden werden oder Parameter und Konfigurationen verändert werden.

Oftmals werden einmal programmierte IPOCS-Applikationen in mehreren Geräten eingesetzt. Dabei wird die Applikation mit einem Zielsystem erstellt, getestet, und parametriert. IPOCS4 erstellt parallel zum Download einer Applikation in das Zielsystem entsprechende APB- und NXE-Dateien. Somit kann die Applikation sehr effektiv mit der Applikationsladefunktion von Netzwerkmanagementtools in viele Knoten kopiert werden.

Steht gerade kein Zielsystem zur Verfügung, kann dieses emuliert werden. Aktivieren Sie zu diesem Zweck die Einstellung ***Offline*** im Dialog ***Einstellungen*** (Abschnitt 3.2). Wenn Sie nun einen Applikationsdownload ausführen, werden die Programmdateien der Applikation (*.APB, *.NXE) erstellt.

5 IPOCS für Inline Control Server

Für Anwendungen, bei denen die Leistungsfähigkeit der Neuron-Chip basierten Geräte nicht ausreicht, können Geräte aus der Familie der Inline Control Server - ICS zum Einsatz kommen. Diese Geräte sind ebenfalls mit IPOCS programmierbar und zeichnen sich aus durch:

- Flexible IO-Konfiguration

  Die Inline Control Server sind Bus-Controller für das modulare Inline-System von Phoenix Contact. Damit können aus dem umfangreichen Katalog der Inline-Klemmen nahezu beliebige IO-Kombinationen zusammengestellt werden. Bis zu 63 Klemmen können an einem ICS betrieben werden; die Kanalanzahl je Klemme variiert je nach Typ zwischen 1 und 32.

- Hohe Performance

  Die ICS-Geräte basieren auf dem 32-Bit-Controller LC-3020 der Firma Loytec. Neben dem entsprechend großzügigerem Speicherausbau steht auch eine deutlich höhere Abarbeitungsgeschwindigkeit zur Verfügung. So können komplexere Applikationen mit kürzeren Zykluszeiten (minimal 10 ms) ausgeführt werden. Damit erschließen sich dem ICS auch neue Anwendungsfelder wie Primäranlagensteuerung und Industrietechnik.

- Erweiterte Netzwerkkommunikation

  Bis zu 4096 Netzwerkvariablen beliebigen Typs können definiert werden, und zwar wahlweise auf dem TP/FT-10- bzw. auf dem IP-Channel (Ethernet – nur bei ICS-852). Die Adresstabelle enthält statt 15 Einträge wie beim Neuron bis zu 65536 Einträge, so dass es keine Probleme mehr bei der Systemintegration gibt

Für die Programmierung der ICS-Geräte mit IPOCS ergeben sich einige Besonderheiten, denen dieses Kapitel gewidmet ist. Detaillierte Informationen finden Sie in der Online-Hilfe der Bausteine

5.1 IO-Zugriff

Der ***IO Configurator*** ist eine Softwarekomponente, mit der eine Inline-Station zusammengestellt und parametriert werden kann. Der Start erfolgt über das Menü ***IPOCS- AddOns > IO Configurator***. Hier wählen Sie die Inline-Klemmen aus einem Katalog aus und stellen daraus eine Inline-Station zusammen. Speziell die Analogklemmen lassen sich umfangreich parametrieren. So kann z.B. für jeden Kanal einer Klemme ein bestimmter Messbereich festgelegt werden. Weiterhin können für alle Kanäle Datenpunktbezeichner angegeben werden.

Der Zugriff auf die Inline-Klemmen, die an das ICS angesteckt werden können, erfolgt kanalweise und typenspezifisch bezogen auf Klemmentyp und den gewählten Messbereich. In der Bibliothek ***Hardware I/O*** finden sich dazu folgende Funktionsbausteine:

Bausteinname	Bedeutung	Signaltyp	Verwendung
DigIn	Digitaleingang	Bit	alle Digitaleingänge
AI	Analog-eingang	Word (% des Messbereiches als SNVT_lev_percent)	alle Analogeingänge außer Temperatureingänge
AmpInF	Stromeingang	Float (A)	Stromeingänge
VoltInF	Spannungs-eingang	Float (V)	Spannungseingänge
TempInF	Temperatur-eingang	Float (°C)	Temperatureingänge
TempInW	Temperatur-eingang	Word (SNVT_temp_p)	Temperatureingänge
ResInF	Widerstands-eingang	Float (Ω)	Widerstandseingänge
DigOut	Digital-ausgang	Bit	alle Digitalausgänge
AO	Analog-ausgang	Word (% des Messbereiches als SNVT_lev_percent)	alle Analogausgänge
AmpOutF	Strom-ausgang	Float (A)	Stromausgänge
VoltOutF	Spannungs-ausgang	Float (V)	Spannungsausgänge
S0Count	Zählereingang	Long (Zählerstand)	S0-Zähler
S0Time	Signal-Zeitmessung	Long (Signallänge in ms)	S0-Zähler

Tabelle 5.1.1: Funktionsbausteine für IO-Zugriff

Alle diese Bausteine haben einen Eingang für den Anschluss eines IO-Kanals. Dabei ist der IO-Kanal ein WORD-Signal, in dessen höherwertigen Byte die Klemmennummer und in dessen niederwertigen Byte die Kanalnummer der Klemme kodiert sind. Mit dem Baustein **ComposeIO** lässt sich die Berechnung der IO-Kanals aus Klemmennummer und Kanalnummer realisieren; der Baustein **DecomposeIO** extrahiert Klemmennummer und Kanalnummer aus einem IO-Kanal.

Bild 5.1.1: IO-Baustein *VoltInF*

Komfortabler ist die Kanalauswahl mit dem Baustein **ParaIO**: Auf Basis der Konfigurationsdaten, die zuvor mit dem IO Configurator erstellt wurden, werden alle verfügbaren Kanäle incl. Typen und Datenpunktbezeichner übersichtlich zur Auswahl gestellt. Besonders interessant: die Synchronisationsfunktion. Wird nachträglich eine Änderung an der Klemmenkonfiguration vorgenommen und verändert sich dadurch die Klemmenreihenfolge, dann können mit dieser Funktion die **ParaIO**-Bausteine an die neue Klemmenkonfiguration angepasst werden. Die Synchronisation basiert dabei auf dem Datenpunktbezeichner. Bei Änderung des Datenpunktbezeichners geht die Parametrierung daher verloren.

Hinweis:
Änderungen an der Klemmenkonfiguration werden erst nach einem erneuten Applikationsdownload im Zielsystem wirksam.

Beim Hochfahren des ICS werden die Inline-Klemmen erkannt und an Hand der Konfigurationsdaten konfiguriert. Wenn die Konfigurationsdaten nicht zu der erkannten Hardware passen, dann leuchtet die IO-LED gelb und mit dem Baustein **TestCfg** kann der zuerst aufgetretene Fehler ausgelesen werden.

Viele Klemmen verfügen über eigene Überwachungsfunktionen wie z.B. Drahtbruch, Messbereichsüberschreitung o.ä. Auf die Ergebnisse dieser Diagnosefunktionen kann mit dem **TestIO**-Baustein zugegriffen werden.

Die optionale Bedien- und Anzeigefunktion des ICS umfasst LEDs, Taster sowie eine 3-stellige 7-Segment-Anzeige. Das ICS verfügt über einen fest programmierten Testmodus, der alle Ausgangskanäle mit den Tastern bedienbar macht. Die Bibliothek **Handbedienung** enthält Bausteine zum Zugriff auf die Bedien- und Anzeigeelemente sowie zum Aktivieren/Deaktivieren des Testmodus und Testtasters.

Simulation

Für den Simulationsbetrieb kann der ICS-Simulator verwendet werden. Diese Anwendung wird über das Menü ***IPOCS-AddOns > Inline ControlServer Simulator*** gestartet und ermöglicht das Setzen von Hardwareeingängen und das Beobachten der Hardwareausgänge des ICS. Der Stationsaufbau wird dabei aus den Daten der Klemmenkonfiguration übernommen.

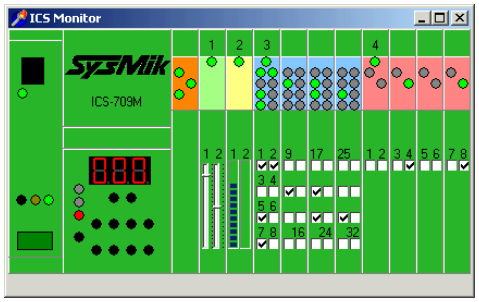


Bild 5.1.2: *ICS Monitor*

RTC / Speicher

Das ICS enthält eine Echtzeituhr (RTC: Real Time Clock) sowie NV-RAM-Speicher, der beliebig oft beschreibbar ist und seinen Inhalt bei Spannungsausfall behält. In der Bibliothek **RTC/Memory** sind Bausteine zusammengefasst, die diese Ausstattungsmerkmale für IPOCS nutzbar machen. So gibt es neben einem RTC-Baustein auch Speicherbausteine mit verschiedenen Basistypen. Damit nach Spannungswiederkehr der gespeicherte Wert gelesen wird, muss sichergestellt werden, dass der Set-Eingang des Speicherbausteins auf 0 steht. Folgendes Beispiel zeigt einen Zähler von Abarbeitungszyklen.

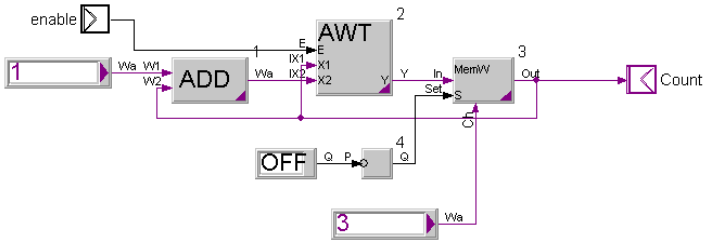


Bild 5.1.3: Applikationsbeispiel der Verwendung von Speicherbausteinen

5.2 Netzwerkvariablen

Alle ICS-Typen verfügen über einen TP/FT-10-Netzwerkanschluss (EIA-709). Das Netzwerkvariableninterface dieses Anschlusses lässt sich frei definieren. Die Bibliothek **NV 709** enthält neben den bekannten Netzwerkvariablenbausteinen zum Zugriff auf Netzwerkvariablen mit Basistyp Word auch gleichwertige Bausteine mit Basistyp Long und Float. Diese Bausteine ermöglichen den Zugriff auf Netzwerkvariablen bzw. Elemente von Netzwerkvariablen mit Basistyp *s32_type* und *float_type*. Der Zugriff auf alle anderen Netzwerkvariablen bzw. Elemente von Netzwerkvariablen erfolgt wie gehabt mit den Word-Bausteinen.

Bei der Definition des Netzwerkvariableninterfaces können entsprechend den Fähigkeiten des Prozessors des ICS

- bis zu 4096 Netzwerkvariablen beliebigen Typs/Richtung
- bis zu 65536 Adresstabelleneinträge und
- bis zu 4096 Aliastabelleneinträge definiert werden.

Bitte beachten Sie vor allem bei der Festlegung der Größe von Adresstabelle und Aliastabelle, dass die Tabellengrößen auch Einfluss auf die Integrationszeit des Gerätes haben. Daher sollten die Tabellen nicht immer die maximale, sondern eine der Anwendung angepasste Größe haben.

Das ICS-852 verfügt neben dem TP/FT-10-Anschluss über einen Ethernet-Anschluss, der die LonWorks-Kommunikation mit dem Protokoll LON/IP (EIA852) ermöglicht. Das entsprechende Netzwerkvariableninterface lässt sich mit Bausteinen der Bibliothek **NV 852** definieren. Über die Verknüpfung der Variablenbausteine von **NV 709** und **NV 852** lassen sich recht einfach Applikationsgateways programmieren.

Die Netzwerkvariablenbausteine unterstützen die Browse-Funktion mit Suchfilter. Über das lokale Menü der Bausteine ***Browse...*** wird ein Dialog geöffnet, in den ein Suchfilter eingegeben werden kann. Der Browser zeigt alle typgleichen Netzwerkvariablenbausteine an, für die gilt:

- der Typname der Netzwerkvariable stimmt mit dem Filterstring überein oder
- der Variablenname (bzw. dessen Anfang) stimmt mit dem Filterstring überein
- der Filterstring entspricht dem Netzwerkvariablenindex.

Beim Importieren oder Einfügen von Netzwerkvariablenbausteinen kann der Anwender entscheiden, ob die von den Bausteinen referenzierten Netzwerkvariablen und die von den Netzwerkvariablen referenzierten LonMark-Objekte in das Netzwerkvariableninterface aufgenommen werden sollen. Dadurch wird es möglich, ganze LonMark-Objekte incl. deren netzwerkseitige Repräsentation zu kopieren.

IPOCS ChannelMapper

Für die effiziente Definition eines puren Mapping zwischen den beiden Netzwerkanschlüssen mit vielen beteiligten Netzwerkvariablen wurde die Komponente **IPOCS ChannelMapper** entwickelt. Sie erlaubt das Zusammenstellen der Netzwerkvariablenzuordnungen in Excel-Tabellen und den nachfolgenden Import. Der **IPOCS ChannelMapper** ist über das Menü ***IPOCS-AddOns > IPOCS ChannelMapper*** startbar. Eine gesonderte Dokumentation finden Sie im IPOCS-Startmenü.

5.3 Verarbeitungsfunktionen

Durch die Leistungsmerkmale des ICS ist eine effektive Float-Verarbeitung möglich; weiterhin steht sehr viel Speicherplatz für Funktionsbausteine zur Verfügung. Aus diesem Grund wurden die von den Neuron-basierten Zielsystemen bekannten Bibliotheken erweitert und weitere Funktionsbibliotheken implementiert. Die von den Neuron-Chip-basierten Zielsystemen bekannten Bibliotheken wurden so weit wie möglich unverändert übernommen, um die Nachnutzung bereits erprobter Programmteile in ICS-Applikationen zu ermöglichen.

Verschiedene arithmetische Standardoperationen (Grundrechenarten) finden sich in den typspezifischen **Arithmetik**bibliotheken. Die Bibliotheken **Auswahl und Vergleich** – ebenfalls nach Typen geordnet – enthalten **Begrenzer, Multiplexer, Schalter** und **Vergleicher**. Standard-Float-Funktionen (Winkelfunktionen u.a.) finden sich in der Bibliothek **Numerik**. Die Bibliothek **Regler** enthält einen **PID-Regler** auf Float-Basis und in der Bibliothek **Zähler** findet sich ein **Vorwärts- / Rückwärtszähler**.

Ablauf-Bausteine

Bei der graphischen Programmierung mit IPOCS werden alle Funktionsbausteine eines Programms in einer festen Reihenfolge zyklisch abgearbeitet. Damit ist der Steuerungsfluss unabhängig von den zu verarbeitenden Daten definiert. Ausnahmen davon können über **Enable**-Bausteine programmiert werden: einzelne Programmteile können in Abhängigkeit von programmierten Bedingungen von der zyklischen Abarbeitung ausgeschlossen werden.

Wesentlich weiter in Richtung von Steuerungsflüssen führt die Funktionsbibliothek **Ablauf-Bausteine**. Sie realisiert Petri-Netze (Schrittsteuerung), die sich aus Schritten (auch als Stellen oder Zustände bezeichnet) bestehen. Schritte sind über Transitionen (Übergänge) miteinander verbunden. Ist die Übergangsbedingung

wahr, dann erfolgt eine Umschaltung auf den nächstfolgenden Schritt. Jeder Schritt besitzt einen Bit-Ausgang, über den z.B. der **Enable**-Baustein eines zugehörigen Makros angesteuert werden kann.

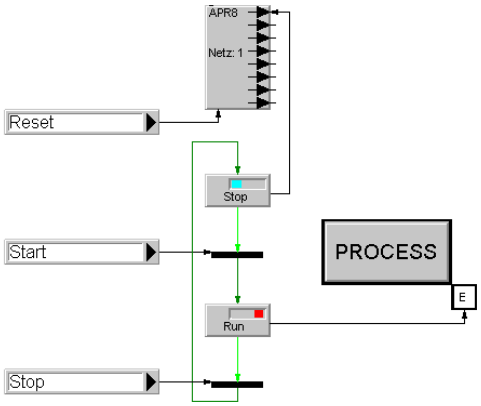


Bild 5.3.1: Applikationsbeispiel Ablauf-Bausteine

Die Funktionsbibliothek **Ablauf-Bausteine** besteht aus folgenden Funktionsbausteinen:

- **Schritt**; Das ist ein Zustand in der Ablaufsteuerung
- **Transition**; repräsentiert einen Übergang von einem Schritt zum nächsten
- **Preset**; ermöglicht das definierte Rücksetzen von Zuständen
- **Verzweigung**; ermöglicht das Aufteilen in voneinander unabhängige Ablaufstränge
- **Zusammenführung**; ermöglicht das Zusammenführen von unabhängigen Ablaufsträngen

Hinweis:
Diese und weitergehende Erläuterungen sind in der Online-Hilfe zu den Bausteinen zu finden. Sie kann über das Kontextmenü oder durch STRG + F1 mit Mauszeiger über dem gewünschten Baustein aufgerufen werden.

Adapter-Bibliotheken: Umwandlung von Datentypen

Neben den normalen Typumwandlungen von Word in Float usw. finden Sie hier auch Zuweisungen (z.B. Word zu Word). Diese Zuweisungen realisieren das Kopieren des Datenwertes an eine neue Speicherstelle. Mit entsprechend rückwärts definierter Bausteinreihenfolge lassen sich auf diese Weise Speicherketten aufbauen.

Weiterhin existieren Umwandlungen in Vektoren: mehrere Daten eines Typs können damit zu Vektoren zusammengefasst werden. Auf die Vektorelemente kann man mit Hilfe der Multiplexer (Bibliothek **Auswahl und Vergleich**) oder mit Split-Bausteinen (Bibliothek **Split-Bausteine**) zugegriffen werden. Folgendes Beispiel realisiert einen Wertespeicher in einem Vektor. Insgesamt 4 Werte werden gespeichert. Wichtig: die Bausteinnummern werden von rechts nach links vergeben.

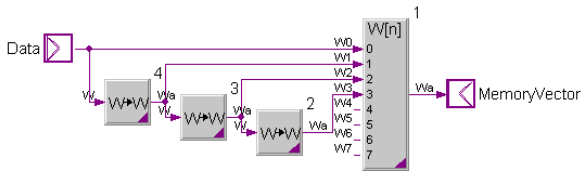


Bild 5.3.2: Applikationsbeispiel Vektor

Die Speicherung erfolgt jeweils bei Werteänderung. Dazu wird der momentane Datenwert mit dem des vorhergehenden Abarbeitungszyklus verglichen. Nur bei Ungleichheit wird das Speichermakro ausgeführt. Der Lesezugriff auf den Wertespeicher erfolgt indiziert über einen Multiplexer.

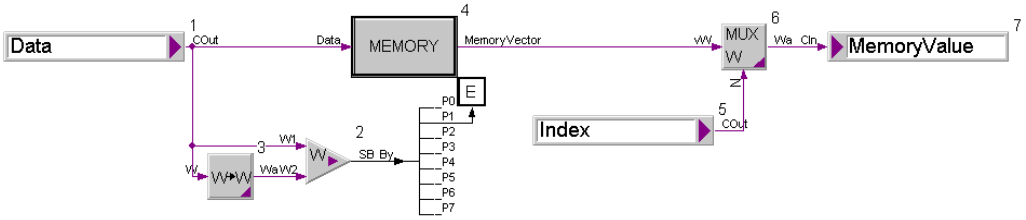


Bild 5.3.3: Applikationsbeispiel Vektor

5.4 BACnet

Es gibt Geräteversionen des ICS mit BACnet Schnittstelle. Dabei werden sowohl BACnet-Server- als auch BACnet-Client-Funktionalitäten angeboten.

Der BACnet-Server stellt BACnet-Objekte in einem BACnet Netzwerk zur Verfügung, auf die ein BACnet-Client (z.B. Leitrechner) lesend und schreibend zugreifen kann. Zur Definition der BACnet-Objekte des ICS mit ihren Properties dient der **BACnet Configurator**. Die Dokumentation „Erst Schritte mit BACnet in IPOCS“ beschreibt den Umgang mit diesem Programm. Folgende Objekttypen werden unterstützt:

- Analog Input / Analog Output / Analog Value
- Binary Input / Binary Output / Binary Value
- Multistate Input / Multistate Output / Multistate Value
- Notification Class
- Device
- Calendar
- Schedule
- Trendlog.

Die Bausteinbibliothek **BACnet Server** stellt für die einzelnen Objekttypen Bausteine zur Verfügung, um den Present_Value der lokalen BACnet-Objekte zu beschreiben bzw. zu lesen. Das Schreiben von kommandierbaren Objekten erfolgt mit Priorität 16. Über einen Anschluss, den alle diese Bausteine besitzen, wird eine Instanznummer übergeben, die das lokale BACnet-Objekt identifiziert, auf welches zugegriffen werden soll. Hilfreich ist hier der Baustein **ParaInst**, mit dem ein Objekt aus der aktuellen BACnet.Konfiguration ausgewählt werden kann.

Die Bibliothek **BACnet Client** enthält Bausteine für den Zugriff auf BACnet-Objekte anderer BACnet-Server. Damit kann die Interkommunikation zwischen verschiedenen BACnet-Automationsstationen realisiert werden. Es stehen folgende Bausteintypen zur Verfügung:

- COV-Übertragung (change of value). Diese Bausteine melden sich bei den angegebenen BACnet-Servern für die COV-Übertragung an, empfangen anschließend die von den Servern verschickten COV-Meldungen und stellen die enthaltenen Daten für die Verarbeitung in der IPOCS-Applikation bereit.
- Read – zyklisches Lesen des Present_Value und der Status_Flags der Objekte
- Write – Schreiben des Present_Value bei Wertänderung bzw. zyklisch

Details zum Umfang der BACnet-Implementation finden Sie im „BACnet Implementation Conformance Statement“.

5.5 Webserver

Alle ICS mit Ethernet-Schnittstelle besitzen einen Webserver, der zahlreiche Funktionalitäten eröffnet:

- feste Konfigurations-Webseiten für die Anzeige von Statusinformationen und Modifizieren von Einstellungen
- Webservices zum Zugriff auf Datenpunkte des ICS
- Anwenderspezifische Webseiten mit aktiven Inhalten.

Detaillierte Informationen dazu lesen Sie bitte in der Dokumentation „ICS Webserver Software“ sowie „Erste Schritte mit dem IPOCS WebDesigner“.

5.6 E-Mail

Alle ICS mit Ethernet-Schnittstelle können bei Bedarf Emails versenden. Dazu wird der Baustein **Send** der Funktionsbibliothek **Email** verwendet. Hier werden in einem Konfigurationsdialog mehrere Emails konfiguriert, wobei neben dem Inhalt auch die Adressen und der Email-Server angegeben werden. Es können auch Platzhalter für Prozessdaten in die Email eingefügt werden, um z.B. aktuelle Messwerte per Email zu verschicken. Weiterhin ist es möglich, den Inhalt von Trendspeichern per Anhang mit einer Email zu versenden. Bitte lesen Sie die Online-Hilfe zu diesem Baustein.

5.7 Systemfunktionen

Alle ICS mit Ethernet-Schnittstelle besitzen jeweils 10 Kalender, 10 Zeitpläne und 40 Trendschreiber, die über den eingebauten Webserver oder alternativ über BACnet (so in der jeweiligen Gerätevariante verfügbar) konfiguriert werden können. Mit den Bausteinen der Bibliothek **System** können die aktuellen Zustände der Kalender, Zeitpläne und Trendschreiber ausgelesen werden, um diese Informationen in der IPOCS-Applikation zu verarbeiten. Der **Trendlog**-Baustein ermöglicht darüber hinaus die Konfiguration und Steuerung der Trendschreiber. Nähere Informationen finden Sie in der Online-Hilfe zu diesen Bausteinen.

5.8 Fidelio

Von der Firma MICROS-FIDELIO stammt die weit verbreitete Hotel-Front-Office-Lösung FIDELIO SUITE. Über diese Software werden Zimmer gebucht, Kunden verwalten, Abrechnungen erstellt u.v.a.m.

Die Fidelio-Schnittstelle des ICS meldet sich bei Aktivierung beim FIDELIO-Server an und erhält eine Information, wann immer ein Gast ein- oder auscheckt. Auf diese Weise kann der Belegungszustand eines jeden Zimmers ermittelt und in der IPOCS-Applikation mit dem Baustein **Occupancy** der Bibliothek **Fidelio** ausgewertet und beliebig verarbeitet werden.

5.9 DALI

Der DALI-Bus erlangt im Bereich der Beleuchtungssteuerung wachsende Bedeutung. Dabei besitzen DALI-fähige Leuchten eine entsprechende Busanschaltung und können über DALI-Befehle gesteuert werden. Die DALI-

Automationsklemmen dienen als Interface zu diesen DALI Netzwerken und sind an jedem ICS betreibbar.

Die Bibliothek **DALI** stellt Bausteine zur Verfügung, mit denen DALI-Leuchten gesteuert werden können. DALI-Klemmen arbeiten als DALI-Master und können bis zu 64 DALI-Slaves (Leuchten) einzeln über Kurzadressen ansteuern. Weiterhin ist die Adressierung von 16 Gruppen sowie Sammeladressierung (alle Slaves werden angesteuert) möglich. Es ergeben sich somit 81 Adressen, die in IPOCS als logische IO-Kanäle der jeweiligen DALI-Klemme betrachtet werden:

- Kanal 1 bis 64: 64 Kurzadressen 0 bis 63
- Kanal 65 bis 80: 16 Gruppenadressen 0 bis 15
- Kanal 81: Sammeladresse.

Die Zusammenstellung einer vollständigen IO-Kanaladresse kann mit den Bausteinen **ParaIO** oder **ComposeIO** der Bibliothek **Hardware I/O** erfolgen.

5.10 EnOcean

Gateways „SRC-Ethernet“ der Firma Thermokon Sensortechnik GmbH setzen Funksignale von bis zu 30 EnOcean-Funksensoren auf Ethernet um. Über Ethernet gelangen die Daten in das Zielsystem. Daher steht diese Funktionalität auch nur bei Verwendung von Zielsystemen mit Ethernetschnittstelle zur Verfügung. Die Bibliothek **EnOcean** enthält Bausteine zum Zugriff auf die empfangenen Sensordaten. Dabei werden die IP-Adresse des Gateways sowie die Instanznummer des jeweiligen Sensors als Parameter übergeben.

5.11 Modbus

Zielsysteme mit Ethernetschnittstelle unterstützen das Feldbusprotokoll Modbus.

Der Modbus TCP Server stellt Modbus-seitig Datenpunkte zur Verfügung, die von anderen Modbus TCP Clients gelesen bzw. beschrieben werden können:

- 10.000 Coils (digitale Ausgänge)
- 10.000 Discrete Inputs (digitale Eingänge)
- 10.000 Holding Register (analoge Ausgänge) und
- 10.000 Input Register (analoge Eingänge).

Die Adressierung der Datenpunkte erfolgt jeweils über die Datenpunktnummern 1 bis 10.000. Für die Einbindung der Modbus-Datenpunkte in die IPOCS-Applikation stehen Bausteine der Bibliothek **Modbus Server** zur Verfügung.

Für den Zugriff auf Datenpunkte anderer Modbus TCP Server dienen Funktionsbausteine der Bibliothek **Modbus Client**. Durch Verwendung eines entsprechenden RS485 Schnittstellenadapters können mit diesen Bausteinen auch Datenpunkte von Modbus RTU-Slaves gelesen und beschrieben werden.

5.12 Zielsystemkerne

Die ICS-Zielsysteme werden mit einem aktuellen Zielsystemkern ausgeliefert. Im Sinne der Produkt-Verbesserung und Funktionserweiterung können neue Zielsystemkerne entstehen, die über das LonWorks Netzwerk in das Zielsystem ladbar sind. Das entsprechende Zielsystemimage trägt die Endung *.dl und kann vom ***Einstellungen***-Dialog des Kommunikationstreibers oder mit dem SysMik LNS-PlugIn **ICS App Loader** in das Zielsystem geladen werden.

Hinweis:
Der Download kann einige Minuten dauern.

Wenn IPOCS das zum Zielsystem passende Kernelimage im Kernel-Verzeichnis des Projektes findet, dann wird beim Download der Applikation ein Applikationsimage (Projektname.al) sowie ein Gesamtimage (Projektname.adl) aus Kernelimage und Applikationsimage erstellt und im Projektverzeichnis abgelegt. Alle Dateitypen können mit dem IPOCS oder dem SysMik LNS-PlugIn **ICS App Loader** verwendet werden, um die Applikation in ICS zu laden.

Hinweis:
Das Applikationsimage funktioniert nur mit dem Kernelimage, welches das Zielsystem zum Zeitpunkt des Applikationsdownloads besaß. Es wird daher sicherheitshalber empfohlen, trotz erhöhtem Zeitbedarf immer das Gesamtimage zu laden.

Wenn das ICS eine Ethernet-Schnittstelle besitzt, kann der Zielsystemkern *.dl auch zeitsparend über den integrierten FTP-Server geladen werden. Verbinden Sie sich mit dem FTP-Server über Nutzernamen „fmwupd-primary“ und Passwort „admin“ und überschreiben die Datei „fmwupd-primary“ im ICS mit der *.dl Datei. Starten Sie das ICS neu, um den neuen Zielsystemkern zu starten.

5.13 Systemdiagnose

Zur Beurteilung der Systemauslastung und Leistungsreserven können zwei Add-Ons verwendet werden. Über das Menü ***IPOCS-AddOns > Ressourcenanzeige...*** kann die Belegung der Speicherbereiche überprüft werden. Dazu muss die Applikation in ein Zielsystem (ggf. PC-Simulation) geladen werden. Über das Menü ***IPOCS-AddOns > Systemdiagnose...*** wird eine Diagnoseoberfläche geöffnet, mit der man

- die Laufzeit der einzelnen Tasks experimentell ermitteln
- eine Systemlastmessung durchführen und
- beliebige Datenbereiche beobachten kann.

5.14 IP-Kommunikationstreiber

ICS-Geräte mit Ethernetanschluss und Zielsystemkern ab Version 55.10 verwenden als Programmierschnittstelle alternativ zu LON auch IP. Vorteil ist eine deutlich höhere Übertragungsgeschwindigkeit (ca. Faktor 10) und damit verbunden kürzere Downloadzeiten. Bei Anmeldung der betroffenen Geräte wird dazu ein zusätzliches Kommunikationstemplate **IP Test** angeboten. In den Einstellungen ist die ***IP-Adresse*** des Zielsystems einzustellen. Die Kommunikation erfolgt dann auf dem Standard-Port 1234. In NAT Umgebungen kann es erforderlich werden, über eine andere Portnummer zu kommunizieren. Dazu kann die IP-Adresse um einen Doppelpunkt als Trennzeichen sowie die Portnummer erweitert werden.

Beim Download eines Zielsystemkerns verwendet der IP-Treiber den FTP-Server des Zielsystems. Ist das Standard-Passwort für den FTP-Server im Zielsystem (admin) verändert worden, dann muss das aktuelle Passwort mit führendem Leerzeichen als Trennzeichen angehängt werden.

Beispiel: **192.168.1.1:4321 ftp_password**

Bei der Definition eines neuen Kommunikationstemplates wird der ***Kommunikationstreiber*** **iconip.dll** ausgewählt, um die schnelle Kommunikation über Ethernet zu nutzen.

Der IP-Treiber unterscheidet sich vom LON-Treiber durch die Kommunikationstechnologie und die damit verbundenen spezifischen Adress-Konfigurationen. Funktionell gibt es keine Unterschiede.

5.15 Programmierbeispiele

Anwendungsbibliotheken

Zum Lieferumfang von IPOCS gehörten mehrere Anwendungsbibliotheken. Diese enthalten fertig programmierte Makros zur Lösung von häufig auftretenden Aufgaben:

- ALLG_FUNKT Allgemeine Funktionen
- GA_FUNKT Gebäudeautomationsfunktionen
- HW_LCS_OMDX Hardwarezugriffe für LCS, OMDX, und MIOX

Einige Bibliotheken enthalten Makros, die Funktionen entsprechend VDI 3813 Blatt 2 abbilden:

- BA_SENSOR Sensorfunktionen
- BA_ACTUATOR Aktorfunktionen
- BA_OPERATOR Bedien- und Anzeigefunktionen
- BA_GENERAL allgemeine Gebäudeautomationsfunktionen
- BA_LIGHTING Lichtsteuerfunktionen
- BA_SHADING Beschattungssteuerungsfunktionen
- BA_HVAC Klimatisierungsfunktionen

Eine Aufstellung der einzelnen Funktionen finden Sie in der **IPOCS Bausteinliste**. Hier ist auch aufgeführt, welche Makros für welche Zielsysteme nutzbar sind.

Über das Menü ***Projekt > Bibliothek laden…*** können die Makrobausteine der Bibliotheken im aktuellen IPOCS-Projekt verwendet werden. **Achtung!** Es wird nur eine Referenz auf die Bibliotheken erzeugt; die Makros werden nicht Bestandteil des Projektes. Die Bausteine besitzen eine online-Hilfe, die über das lokale Menü in IPOCS oder über die **IPOCS Bausteinliste** aufgerufen werden kann.

Die Anwendungsbibliotheken sind ganz normale IPOCS-Projekte, die im Verzeichnis …\IPOCS\Lib\Makro abgelegt sind. Sie können diese Projekte öffnen, Teile kopieren oder exportieren und die Makros Ihren speziellen Bedürfnissen anpassen. **Achtung!** Änderungen an einer Bibliothek haben Auswirkungen auf alle Projekte, die auf diese Bibliothek referenzieren.

LonMark-Objekte

Um die Erstellung von LonMark-konformen Applikationen zu vereinfachen, werden LonMark-Objekte als Makros zur Verfügung gestellt. Sie können sich aus diesen Makros eine Applikation komfortabel zusammenstellen. Lesen Sie dazu bitte die Dokumentation „Nutzung der LonMark Objekte“.

DALI Festapplikationen

In der Sammlung der Programmierbeispiele im Verzeichnis „Sample“ der IPOCS-Installation befinden sich Applikationen für ICS mit DALI-Klemmen zur LonMark - Lichtsteuerung. Eine detaillierte Beschreibung dazu ist über das Startmenü … IPOCS - „Dokumentation“ – „ICS DALI Applikationen zur Lichtsteuerung“ erreichbar.

6 Ausblick: Was ist noch möglich?

Sie haben nun die grundlegenden Funktionen von IPOCS-4.1 kennengelernt. Hier werden nun weitere Eigenschaften und Funktionen kurz aufgezeigt.

Hinweis:
Viele Fragen kann Ihnen die in IPOCS-4.1 integrierte Hilfe beantworten. Diese können Sie über das Menü ***?*** aufrufen.

Zielsystemtemplates

Verwenden Sie für Ihre Projekte eigene Zielsystemtemplates mit unikaten, deskriptiven Namen (**LON-Test** ist nur für Testzwecke gedacht). Klicken Sie dazu im Fenster ***Zielsystem anmelden*** mit der rechten Maustaste in die Liste der Zielsysteme. In dem sich daraufhin öffnenden lokalen Menü wählen Sie ***Template hinzufügen ...*** und tragen im Dialog Template bearbeiten den deskriptiven ***Name***n (das ist dann gleichzeitig der Name des beim Anmelden automatisch startenden Kommunikationsservers), eine ***Beschreibung*** (so wird das Template später in der Liste der Templates angezeigt) sowie ***Serverklasse*** **IO32NS** und ***Kommunikationstreiber*** **iconlon.dll** ein. Für die Kommunikation über Ethernet verwenden Sie bitte den ***Kommunikationstreiber*** **intracom.dll**.

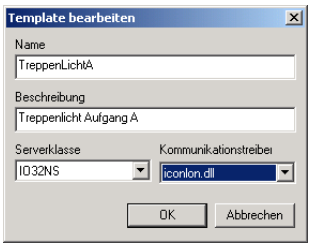


Bild 6.1: Nutzerdefiniertes Zielsystemtemplate

Konfiguration

Das hier vorgestellte Beispiel besteht aus nur einer Instanz eines Programmbausteins. Dies ist die Standardkonfiguration, die beim Erstellen eines neuen Projektes automatisch aktiviert wird. Die Applikation kann aber aus bis zu 15 Instanzen verschiedener Programmbausteine (Tasks) bestehen.

In der Konfiguration (***Bearbeiten > Konfiguration***) werden die einzelnen Tasks angelegt. Weiterhin wird Priorität der Tasks festgelegt (***Bearbeiten > Baustein-Reihenfolge***), wodurch der Programmablauf entscheidend beeinflusst wird. Eine umfangreiche Task hoher Priorität (***Bearbeiten > Baustein-Reihenfolge***) kann durchaus die Abarbeitung einer niederpriorisierten Task verzögern bzw. verhindern.

Beim Eintritt in die Inbetriebnahme (***Inbetriebnahme > Eintritt***) können über das lokale Menü die Zykluszeiten der Tasks festgelegt werden.

Eine Applikation kann mehrere Instanzen eines Programmbausteins enthalten. Zur Unterscheidung der Instanzen können in der Inbetriebnahme (***Inbetriebnahme > Eintritt***) über die lokalen Menüs Instanznamen für die Tasks vergeben werden. Weiterhin sollten hier Anpassungen der Parameter vorgenommen werden: Beim Anlegen einer Instanz eines Programmbausteins werden automatisch die Parametereinstellungen des Programmbausteins in die Instanz kopiert. Es ist aber beispielsweise nicht sinnvoll, zwei Treppenlichtsteuerungen auf einen Relaisausgang wirken zu lassen. Hier sollte in den Instanzen die Kanalnummer des Relaisausganges verändert werden.

Kommunikation zwischen Programmbausteininstanzen

Die konfigurierten Tasks können Daten untereinander austauschen. Dazu werden in den Programmbausteinen **Input** bzw. **Output** Funktionsbausteine der Bibliothek **Standard** angelegt (Der Typ wird über einen Konfigurationsdialog angegeben) und mit den anderen Funktionsbausteinen innerhalb der Programmbausteine verbunden. Über das Menü ***Strukturbaustein > Design*** wird ein Fenster geöffnet, in dem per Drag & Drop-Operation alle Inputs und Outputs auf dem Programmbausteinrahmen platziert und damit die Anschlussstellen definiert werden.

In der Konfiguration verfügen nun die entsprechenden Tasks über die definierten Anschlussstellen. Über Verbindungen zwischen diesen Anschlussstellen erfolgt die Kommunikation zwischen den Tasks.

Aktivieren / Deaktivieren von Programmbausteininstanzen

Sie können die Abarbeitung einzelner Tasks steuern, indem Sie in der Konfiguration einen Funktionsbaustein **Enable-SB** der Bibliothek **Standard** direkt auf der entsprechenden Task platzieren. Er heftet sich an die untere rechte Ecke der Task und stellt einen Biteingang zur Verfügung. Die Task wird nur abgearbeitet, wenn dieses Signal auf High steht.

Makrobausteine

Über den Menüpunkt ***Bearbeiten > Makrobaustein*** können Makrobausteine programmiert werden, in denen häufig wiederkehrende Bausteinkombinationen zusammengefasst werden. Die Makrobausteine enthalten **Input** bzw. **Output** Funktionsbausteine der Bibliothek **Standard**, die Anschlussstellen des Makrobausteins darstellen (***Strukturbaustein > Design***). Instanzen von Makrobausteinen können in Programmbausteine eingefügt werden.

Zielsystemkerne

Es kann zu einzelnen Zielsystemen mehrere Zielsystemkerne (IPOCS-Firmware + Bausteine) geben. Diese entstehen durch ständige Weiterentwicklung der Bausteine oder durch ganz spezielle Ausrichtung der Bibliotheken. In jedem Projektverzeichnis existiert ein Unterverzeichnis **Kernel**, in dem die jeweiligen Zielsystemkerne (***.nxe** und ***.apb**) enthalten sind. Wenn Sie also einen neuen Zielsystemkern in Ihr Zielsystem laden möchten, können Sie dies mit Hilfe dieser Dateien (Download über gängige Netzwerkmanagementtools) tun.

Hinweis:
Ein Zielsystemkern ist immer hardwarespezifisch: Ein Zielsystemkern für LCS-410 wird nicht auf einem Zielsystem LCS-310 laufen.

Bibliotheks-Update

Beim Anlegen eines neuen Projektes definieren Sie über die Auswahl des Zielsystems auch die zu ladenden Bibliotheken. Wollen Sie das Projekt später zur Programmierung eines anderen Zielsystems verwenden, dann müssen Sie ggf. andere Bibliotheken laden. Um eine neue Bibliothek in IPOCS-4.1 zu laden, wählen Sie den Menüpunkt ***Projekt > Bibliothek laden***. Das Löschen von Bibliotheken erfolgt auch unter dem Menüpunkt ***Projekt > Bibliothek löschen***. Sie können eine vorhandene Bibliothek aber auch ersetzen, indem Sie den Befehl ***Projekt > Bibliothek ersetzen*** benutzen.

Globale Variablen

In vielen Applikationen existieren Daten, auf die verschiedene Makros und Programmbausteine zugreifen müssen. Speziell in stark strukturierten Applikationen

führt dies zu einem aufwändig zu erstellendem und zu pflegendem Gewirr von Verbindungslinien zwischen den Strukturbausteinen.**Browser**

In der Bibliothek **Globale Variablen** sind Bausteine enthalten, um globale Variablen zu definieren und zu beschreiben (**Wert schreiben**) sowie globale Variablen zu lesen (**Wert lesen**). Somit kann von jeder Stelle der Applikation aus einfach auf globale Daten zugegriffen werden.

Globale Variable erzeugen im Zielsystem keinerlei Code – sie sind vielmehr eine alternative Darstellungsform von Verbindungen zwischen Funktionsbausteinen.

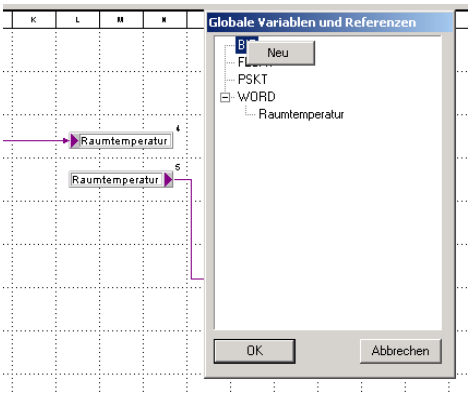


Bild 6.2: Globale Variablen

Browser

Gerade in komplexen, stark strukturierten Projekten ist es aufwändig, einzelne Bausteine manuell zu suchen. Im lokalen Menü jedes Funktionsbausteines finden Sie den Menüpunkt ***Browse...***, der den ***Baustein-Browser*** startet. Nach dem optionalen Modifizieren der Suchkriterien werden alle Funktionsbausteine, die die Suchkriterien erfüllen, tabellarisch angezeigt. Ein Doppelklick auf den Bausteinnamen in einer Zeile aktiviert das entsprechende Arbeitsblatt und markiert den Baustein. Die Suchkriterien sind abhängig vom Funktionsbaustein, von dem aus der Browser aufgerufen wurde. Die Funktionsbausteine der Bibliothek **Globale Variablen** erlauben das Browsen nach Namen der globalen Variablen sowie Art des Zugriffs (Schreiben / Lesen).

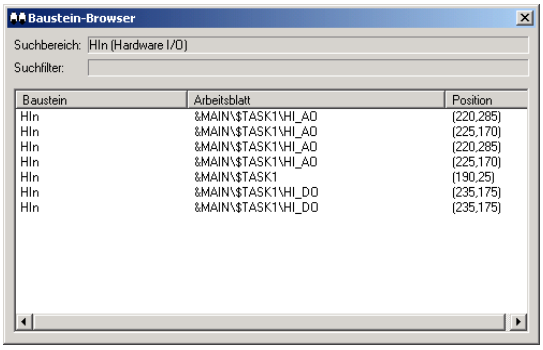


Bild 6.3: Baustein-Browser

Passt die Applikation in das Zielsystem?

Über das Menü ***IPOCS-AddOns > Ressourcenanzeige...*** wird ein Dialog geöffnet, in dem die Ressourcenauslastung des Zielsystems dargestellt wird. Dabei kann man zu jedem Speicherbereich den im Zielsystem verfügbaren und den durch die Applikation belegten Speicher ansehen. Voraussetzung dafür ist, dass die

Applikation in ein Zielsystem (ggf. auch Simulation) geladen wurde. Nur so stehen die darzustellenden Informationen zur Verfügung.

Mit dieser Funktion lässt sich schon frühzeitig der Speicherbedarf einer Applikation ermitteln und ggf. optimieren.

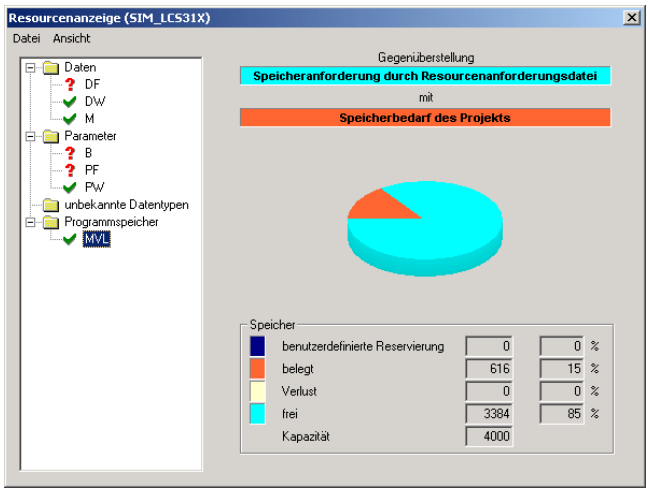


Bild 6.4: Ressourcenanzeige

IEC 61131

Mit der IEC 61131-3 wurden herstellerneutrale Sprachen zur Programmierung von Steuerungen genormt. Diese Sprachen sind typische Ausbildungsinhalte in Lehre und Studium und damit weit verbreitet. Die Programmierung mit IPOCS entspricht im Wesentlichen der in der IEC 61131 definierten Sprache FUP (Funktionsplan). Unterschiede bestehen jedoch in Funktion und Namensgebung der einzelnen Funktionsbausteine. Um für IEC 61131-erfahrene Programmierer die Arbeit zu erleichtern, wurden Makros geschaffen, die den aus der IEC 61131 bekannten Funktionsbausteinen gleichen. Die Makros sind in Makrobibliotheken zusammengefasst:

Bibliothek	Beschreibung	Inhalt
IEC Arith	arithmetische Grundfunktionen	ADD, SUB, MUL, DIV, MOD, ABS, MIN, MAX, SEL
IEC Comp	Vergleichsoperatoren	GT, GE, EQ, LE, LT, NE
IEC Convert	Typ-Konvertierungen	BOOL_TO_INT, INT_TO_BOOL, INT_TO_REAL, TRUNC
IEC Count	Zählerbausteine	CTU, CTD, CTUD
IEC Flipflop	Bistabile Funktionsbausteine	RS, SR
IEC Logic	logische Operatoren	AND, OR, XOR, NOT
IEC Timer	Timer – die Angabe der Zeiten erfolgt immer in 100 ms-Schritten (Eingangswert von 10 entspricht 1s)	TON, TOF, TP, BLINK
IEC Trigger	Flankenerkennung	F_TRIG, R_TRIG

Tabelle 6.1: IEC-Makro-Bibliotheken

Auf eine detaillierte Beschreibung der Makros wird verzichtet, da diese Makrobibliotheken ausdrücklich für Anwender erstellt wurden, die über Erfahrungen mit IEC-konformen Sprachen verfügen.

Um die Makrobibliotheken in einem Projekt nutzen zu können, müssen diese in das Projekt geladen werden. Öffnen Sie dazu das entsprechende Projekt und laden Sie über das Menü ***Projekt > Bibliothek laden...*** die benötigten Makro-Bibliotheken. Die Bibliotheken bzw. die darin enthaltenen Makros stehen nun in der Lise der Bibliotheken zur Verwendung bereit.

Um die Bezeichnungen der Anschlüsse der Makros zu aktivieren, wählen Sie bitte den Menüpunkt ***Optionen > Anschlüsse***.

Die Makrobibliotheken sind IPOCS-Projekte, die im Verzeichnis ...IPOCS41\LIB\MAKRO abgelegt sind. Sie lassen sich hier auch öffnen und ggf. erweitern. Bitte beachten Sie, dass Änderungen in diesen Makrobibliotheken Einfluss auf alle Projekte haben, die auf diese Makros zugreifen.

LNS-Plug-Ins

Plug-Ins erleichtern die Inbetriebnahme von Geräten in LonWorks-Netzwerken. Mit dem PlugIn-Creator steht ein Werkzeug zur Verfügung, mit dem applikationsspezifische LNS-Plug-Ins für IPOCS-programmierte Geräte erstellt werden können. Die Netzwerkvariablen der Applikation bzw. Teile davon werden dabei in einer frei definierbaren Hierarchie angeordnet und mit Zusatzinformationen (Einheit, Minimalwert, Maximalwert, Beschreibungstext etc.) versehen. Das so generierte Plug-In ermöglicht das einfache Parametrieren sowie das Beobachten der Geräte. Eine genaue Handlungsanleitung ist im Dokument „Erste Schritte mit dem IPOCS Device PlugIn“ enthalten.

Display DMTX programmieren

Neben den Steuergeräten steht auch ein Display zur Verfügung, welches mit IPOCS grafisch programmiert werden kann. Dadurch erschließen sich völlig neue Aufgabenklassen. Die zur Verfügung stehenden Funktionalitäten umfassen u.a. die Ausgabe von Zeichenketten und Prozessdaten sowie das Editieren von Parameterwerten. Mit der Hilfe von Petrinetzen können Menüstrukturen realisiert werden. Das Dokument „Erste Schritte mit dem DMTX“ enthält eine detaillierte Anleitung zur Programmierung von Displays.

Debug

Will man die Programmabarbeitung einzelner Programmbestandteile genau unter die Lupe nehmen, kann mit folgendem Konstrukt ein Einzel-Zyklusbetrieb realisiert werden. Dabei kann man über eine 0-1-Flanke der globalen Variablen *Step* jeweils einen Zyklus des Makros *Debug* abarbeiten und sich danach in aller Ruhe die Signale ansehen.

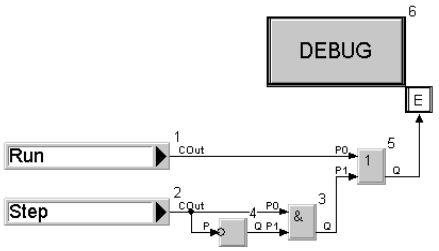


Bild 6.5: Applikationsbeispiel zum Debuggen von Programmteilen

7 Literatur

[1] IPOCS-Online-Hilfe

[2] IPOCS Bausteinliste, SysMik GmbH Dresden

[3] Handbuch: Erste Schritte mit dem PlugIn Creator, SysMik GmbH Dresden

[4] Handbuch: Erste Schritte mit dem IPOCS Device PlugIn, SysMik GmbH Dresden

[5] Handbuch: Erste Schritte mit dem IO Configurator, SysMik GmbH Dresden

[6] Handbuch: Erste Schritte mit BACnet in IPOCS, SysMik GmbH Dresden

[7] BACnet Protokoll Implementation Conformance Statement , SysMik GmbH Dresden

[8] Handbuch: Nutzung der LonMark Objekte, SysMik GmbH Dresden

[9] Handbuch: Erste Schritte mit DMTX, SysMik GmbH Dresden

[10] Handbuch: ICS Webserver Software, SysMik GmbH Dresden

[11] Handbuch: Erste Schritte mit dem IPOCS WebDesigner, SysMik GmbH Dresden

[12] Handbuch: ICS DALI Configurator, SysMik GmbH Dresden

[13] Handbuch: ICS DALI Applikationen zur Lichtsteuerung, SysMik GmbH Dresden

[14] www.sysmik.de

[15] SNVT Master list, LonMark

[16] LonMark Application-Layer Interoperability Guidelines, LonMark

[17] www.lonmark.org

[18] Handbuch: Erste Schritte mit dem LON Configurator, SysMik GmbH Dresden